



**CHALMERS**



**GÖTEBORGS UNIVERSITET**

# **Styrsystem för en övertons- strålningskälla optimerad för attosekundsspektroskopi**

Kandidatarbete inom fysik

Ludvig Ivehag  
Adnan Jordamovic  
Daniel Mannerskog

**INSTITUTIONEN FÖR FYSIK**

---

UNIVERSITY OF GOTHENBURG  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Gothenburg, Sweden 2021



# Styrsystem för en övertons-strålningskälla optimerad för attosekundsspektroskopi

Kandidatarbete inom fysik

Grundnivå  
180 hp

3 juni 2021

Ludvig Ivehag, ludde.ivehag@gmail.com  
Adnan Jordamovic, adnjor@gmail.com  
Daniel Mannerskog, daniel.mannerskog@gmail.com

Handledare:  
Raimund Feifel  
Richard Squibb  
Hélène Coudert-Alteirac  
Emelie Olsson



**CHALMERS**



**GÖTEBORGS UNIVERSITET**



## Författarnas tack

Ett stort tack till Raimund Feifel, Richard Squibb, Hélène Coudert-Alteirac och Emelie Olsson för den omfattande hjälp vi fått av er i detta arbete.

Tack Raimund för möjligheten till att utföra detta projekt.

Tack Richard för hjälpen med uppställningen och svar på frågor om hårdvaran som användes.

Tack Hélène för alla svar på frågor om HHG.

Tack Emelie Olsson för att labbdatorns nätverksuppkoppling fungerade.



## Abstract

A control system was developed for an HHG beamline at Attohallen, which is currently being built at the University of Gothenburg. The system was programmed in *National Instruments's* graphical programming language LabVIEW. The control system is made up of four different parts, vacuum pumps, pressure gauges, cameras and piezomotors. The result is a functioning pressure control system with adaptive features for easy navigation and a functioning communication system with camera and controls of frame rate and exposure time.

**Key words:** Control system, LabVIEW, High-order Harmonic Generation, Vacuum, Attohallen.

## Sammanfattning

Ett styrsystem utvecklades för den planerade HHG-uppställning som för närvarande byggs vid Göteborgs universitet, Attohallen. Systemet utvecklades i *National Instruments* grafiska programmeringsspråk LabVIEW. Styrsystemet består av fyra delar, vakuumpumpar, tryckmätare, kameror och piezomotorer. Resultatet är ett fungerande tryckprogram med anpassningsbara funktioner för enkel navigering samt en fungerande kommunikation med kameran och kontroll över bildfrekvens samt exponeringstid.

**Nyckelord:** Kontrollsystem, LabVIEW, Hög övertongsgenerering, Vakuum, Attohallen.





# Innehåll

<b>Akronymer</b>	<b>i</b>
<b>1 Introduktion</b>	<b>1</b>
<b>2 Bakgrund och motivering</b>	<b>2</b>
2.1 LabVIEW . . . . .	2
2.2 HHG-tekniken . . . . .	3
2.3 Inverkan av luft på IR- och XUV-pulser . . . . .	6
2.3.1 Pulsbreddning genom dispersionen för IR i luft . . . . .	6
2.3.2 Transmission av XUV i luft . . . . .	8
2.4 Syfte och mål för arbetet . . . . .	10
<b>3 Hårdvara</b>	<b>11</b>
3.1 Vakuumpumpar . . . . .	11
3.1.1 Rotorpumpar . . . . .	11
3.1.2 Turbopumpar . . . . .	12
3.2 Tryckmätare . . . . .	12
3.2.1 Piranimätare . . . . .	13
3.2.2 Kallkatodjonisationsmätare . . . . .	13
3.3 Piezomotorer . . . . .	14
3.4 Hårdvarukommunikation . . . . .	15
3.4.1 Kommunikation med tryckmätare och turbopumpar . . . . .	15
3.4.2 Kommunikation med kameror . . . . .	16
<b>4 Metoder och kommunikationsprotokoll</b>	<b>16</b>
4.1 Tryckreglering . . . . .	16
4.1.1 Kommunikation med tryckmätare . . . . .	17
4.1.2 Kommunikation med turbopumpar . . . . .	18

4.2	Optik . . . . .	19
4.2.1	Struktur för att hämta bildinfo via LabVIEW . . . . .	20
4.2.2	Beräkning av laserns position i en bild . . . . .	20
<b>5</b>	<b>Resultat &amp; Diskussion</b>	<b>21</b>
5.1	Tryckreglering . . . . .	22
5.2	Optik . . . . .	28
<b>6</b>	<b>Slutsats</b>	<b>30</b>
<b>7</b>	<b>Referenser</b>	<b>31</b>
<b>A</b>	<b>Textrepresentationer av numeriska värden</b>	<b>I</b>
<b>B</b>	<b>Fullständig kod</b>	<b>II</b>



## Akronymer

CCIG	Kallkatodjonisationsmätare (Cold Cathode Ionization Gauge)
COM	Kommunikation (Communication)
CPA	Pulsförstärkning (Chirped Pulse Amplification)
HHG	Hög övertongsgenerering (High-order Harmonic Generation)
HV	Högt vakuum (High Vacuum)
IR	Infraröd (Infrared)
LV	Lågt vakuum (Low Vacuum)
TMP	Turbomolekylärpump (Turbo Molecular Pump)
UHV	Ultrahögt vakuum (Ultra-High Vacuum)
XUV	Extremt ultraviolett (Extreme Ultraviolet)

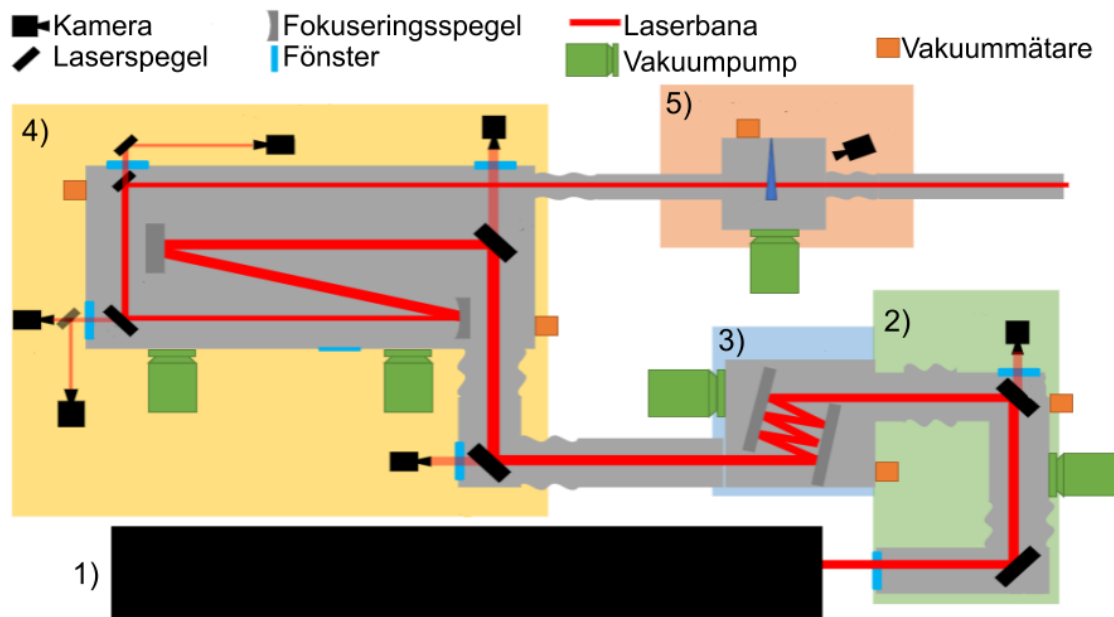


# 1 Introduktion

Atomerna finns överallt runtom oss, i luften vi andas, vattnet vi dricker och marken vi står på - överallt kan vi finna dem. Av den anledningen finns det ett värde i att samla mer kunskap om atomers egenskaper i syfte att utveckla en bättre förståelse för vår omgivning. Hittills har kunskapen om atomer gett oss möjligheten till att rädda liv genom nya läkemedel, den har gett oss utveckling inom nya material och mycket mer. Kunskapen om atomen har kommit en lång väg sedan upptäckten av dess existens men det finns fortfarande mycket kvar att utforska, vilket kräver att man går djupare in i fysikens värld för att få en tydligare bild av hur atomen ser ut. En förutsättning för att bygga upp bilden av atomer är att det utförs fysikaliska experiment, där närmare undersökningar av olika subatomära och interatomära processer studeras.

Sådana fysikaliska experiment är dock långt från triviala att utforma, delvis på grund av att fysikaliska förändringar på atomär nivå kan ske under mycket korta tidsintervall. En relativt ny teknik som används idag för att undersöka dessa snabba förändringar är hög övertonggenerering, förkortat HHG (*high-order harmonic generation*). I denna teknik genereras högfrekventa elektromagnetiska pulser med otroligt korta tidsintervall på attosekundsskala ( $10^{-18}$  s). Tack vare att pulserna är korta kan HHG-uppställningar användas för att undersöka ultrasnabb dynamik inuti atomer. En sådan undersökning har utförts av C.L. Arnold med flera [1].

I dagsläget byggs Attohallen vid Göteborgs universitet, där en uppställning av HHG-tekniken ska användas. I figur 1 visas en schematisk bild på hur HHG-uppställningen kommer att se ut. I figuren illustreras en laserstråles väg genom de olika vakuumkammarna där speglar, kameror, pumpar och tryckmätare kommer vara placerade. Varje kammare kommer ha minst en uppsättning av vakuumpumpar var och lika många tryckmätare som senare kommer bli automatiserat av vårt LabVIEW-program. Lasern leds med hjälp av speglar som kommer vara placerade längs kamrarnas väggar. Bakom varje spegel kommer det att finnas en kamera som bestämmer laserns position på spegeln vilket möjliggör automatisering för att leda lasern genom kamrarna.



Figur 1: Schematisk bild över HHG-uppställningen som i nuläget byggs vid Göteborgs universitet. De gråa ytorna i bilden illustrerar vakuum. 1) Laser. 2) Transport. 3) Kompressorkammare. 4) Fokuseringskammare. 5) Generationskammare.

För att HHG-tekniken ska fungera krävs det att laserstrålen propagerar i mycket lågt tryck samt att laserstrålen är väl riktad och träffar en specifik punkt i generationskammaren, i figur 1. Med anledning av detta är målet med arbetet, presenterat i denna rapport, att utveckla styrsystem för tryckreglering och riktning av laserstrålen. De utvecklade styrsystemen ska automatisera vakuumbildning i kamrarna samt leda lasern genom uppställningen. I följande avsnitt motiveras nödvändigheten av de utvecklade styrsystemen, följt av en beskrivning av hårdvaran och hur systemen kontrolleras och slutligen presenteras funktionaliteten för de färdigställda systemen.

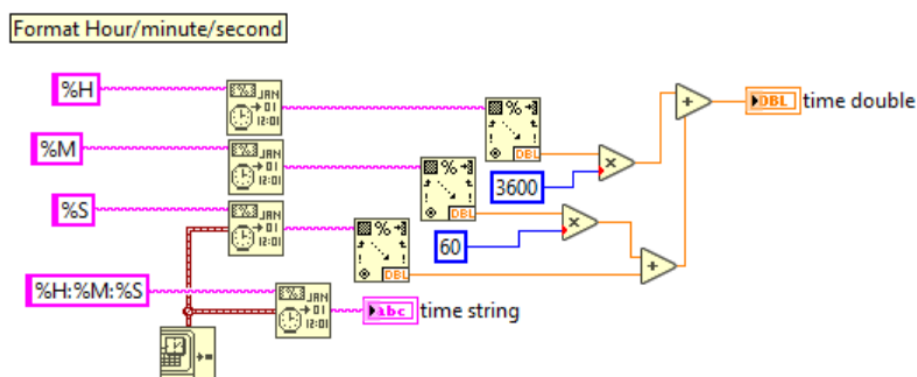
## 2 Bakgrund och motivering

Anledningen till att styrsystem för tryckreglering samt riktning av laserstrålen behövs är för att minimera energiförluster i HHG-uppställningen. Nedan följer en mera detaljerad motivering av detta påstående samt en översiktlig bild över hur styrsystemen implementerats för att uppfylla detta mål. Inledningsvis ges en kort förklaring av LabVIEW.

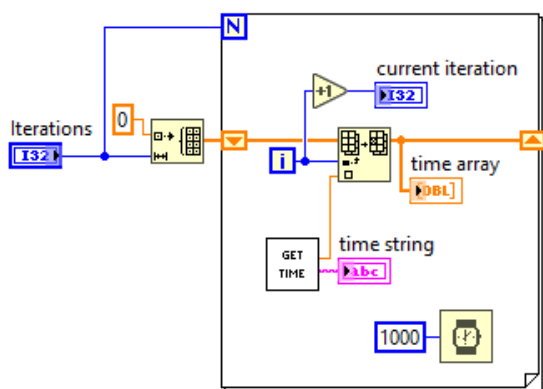
### 2.1 LabVIEW

Styrsystem utvecklades i LabVIEW, ett grafiskt programmeringsspråk utvecklat av *National Instruments*. LabVIEW används till att automatisera labbutrustning genom att kommunicera med hårdvara.

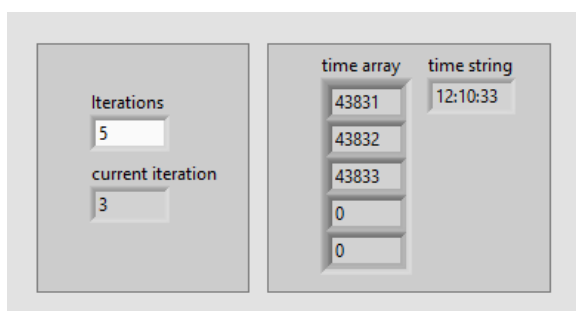
Programmeringen i LabVIEW utförs genom konstruktion av blockdiagram där data skickas mellan olika operatörer som exempelvis kommunicerar med instrument eller utför matematiska beräkningar. I figur 2a visas ett blockdiagram för *getTime*, en så kallad sub:VI, vilken hämtar datorns interna tid och returnerar den som en textsträng samt antal sekunder, med referens 00:00 (HH:MM). En sub:VI kan kallas av andra VI, vilket visas i figur 2b där *getTime* används för att placera den uppmätta tiden i en vektor varje sekund. Tillhörande samtliga blockdiagram finns även en frontpanel, figur 2c. Det är i frontpanelen användaren styr programmet, i figur 2c kan exempelvis antalet iterationer ändras. Dessutom kan data från blockdiagrammet visas i frontpanelen.



(a) Blockdiagram för *getTime*.



(b) Exempel på användning av *getTime*.



(c) Frontpanel för den VI som visas i (b).

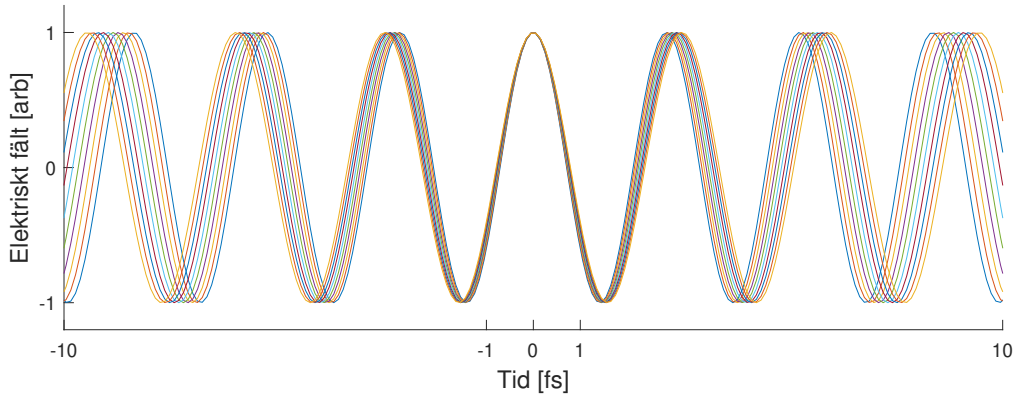
Figur 2: Ett blockdiagram till en sub:VI, *getTime*, samt hur denna sub:VI kan användas i en annan VI.

## 2.2 HHG-tekniken

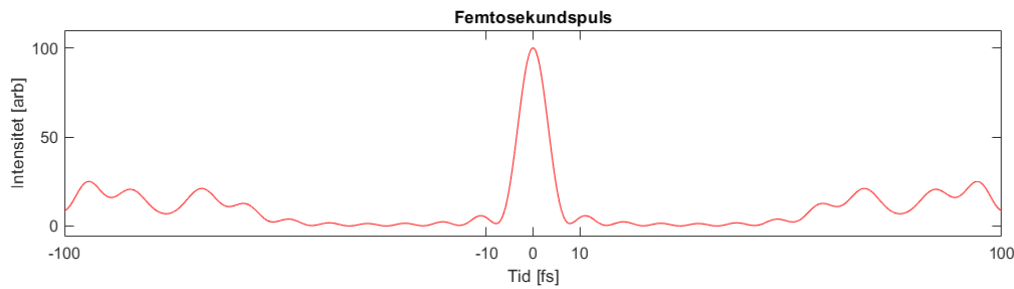
HHG-tekniken används för att skapa attosekundspulser av elektromagnetisk strålning, där våglängdskomponenterna som bygger upp pulsen är i det extrem-ultraviolette området, XUV. Genereringen av dessa pulser kan delas in i två huvudsakliga steg. I första steget skapas en femtosekundspuls med hjälp av en laser. Laserpulsens används i det andra steget till att först jonisera gaspartiklar och därefter återförena de skapade jonerna med de utslagna elektronerna. Vid denna återförening sänds XUV-pulser ut.



Femtosekundspulser skapas i en laserprocess där ljusvågor med frekvenser i det infraröda (IR) spektrat superpositioneras. Det krävs alltså att icke-monokromatiskt laserljus skapas, vilket typiskt görs med hjälp av en titandopad safirkristall [2]. Genom att synkronisera vågkomponenternas fas, när de skickas ut vid  $t = 0$ , skapas en kort puls med en hög effekt vid denna tid och betydligt lägre vid andra tidpunkter. Idén illustreras i figur 3, där tio monokromatiska vågkomponenter, figur 3a, med frekvenser mellan 850 nm och 950 nm summeras för att skapa en enskild puls, figur 3b, med en bredd i storleksordningen 10 fs. Genereringen av denna laser sker vid 'Laser', markering 1 i figur 1.



(a) Infraröda ljusvågor som bygger upp pulsen i (b).



(b) Femtosekundspulsens intensitet vid summeringen av ljusvågorna i (a).

Figur 3: Summering av monokromatiska vågor för att skapa en kort puls. Notera att pulsen har sitt största värde vid  $t = 0$  där alla enskilda vågor ligger i fas. I övre bilden visas den elektriska fältkomponenten  $E_\lambda$  för respektive våg och i undre bilden den totala effekten  $\propto (\sum_\lambda E_\lambda)^2$ .

Beroende på fysikaliska begränsningar i den titandopade safirkristallen skickas enbart våglängder i ett begränsat spektrum ut. I en experimentuppställning använd av J. Solis med flera [3] genererades ett kontinuerligt spektrum med våglängder mellan 770 nm och 800 nm vilket resulterade i en puls på 43 fs. Med ett större spektrum kan kortare pulser skapas vilket kan förklaras med att en summering av vågor med ett stort antal våglängder, som ligger i fas vid  $t = 0$ , är likt en ideal Fouriertransform av en deltafunktion. Effekten beskrivs ofta med tid-bandbreddsprodukten, produkten mellan ljuspulsens varaktighet och dess spektralbredd. En kort varaktighet hos en puls ger ett brett frekvensspektrum och en lång varaktighet ger ett smalt frekvensspektrum. Matematiskt beskrivs detta enligt

$$\Delta t \Delta \omega = A, \quad (1)$$

där  $\Delta t$  och  $\Delta\omega$  är standardavvikelsen för pulsen i tidsdomänen respektive vinkelfrekvensdomänen.  $A$  antar olika värden beroende på pulsens form.

De genererade laserpulserna har i detta steg för lågt energiinnehåll för att generera attosekundspulser. CPA, *Chirped Pulse Amplification*, används därför för att förstärka pulsen. Tekniken bygger på att pulsen sträcks ut i tidsdomänen. Detta genom att fasförskjuta de olika våglängdskomponenterna relativt varandra, vilket bevarar den totala energin i pulsen, men minskar den högsta effekten. Efter utsträckningen förstärks energin i pulsen varefter pulsen återigen komprimeras, markering 3 i figur 1 [4]. Anledningen till att pulsen sträcks ut innan förstärkningen är för att inte skada kristallen som förstärker pulsen [2].

I fokuseringskammaren, markering 4 i figur 1, fokuseras laserstrålen för att maximera energin som passerar genom en liten tvärsnittsytta i generationskammaren innehållande en gas, markering 5.

Genereringen av attosekundspulser sker vid markering 5 och kan beskrivas genom att betrakta vad som sker före, under och efter en femtosekundspuls belyst gasen. Gasen som belyses av pulsen är typiskt en ädelgas [5], vilket innebär att gasen kan beskrivas som en samling enskilda atomer.

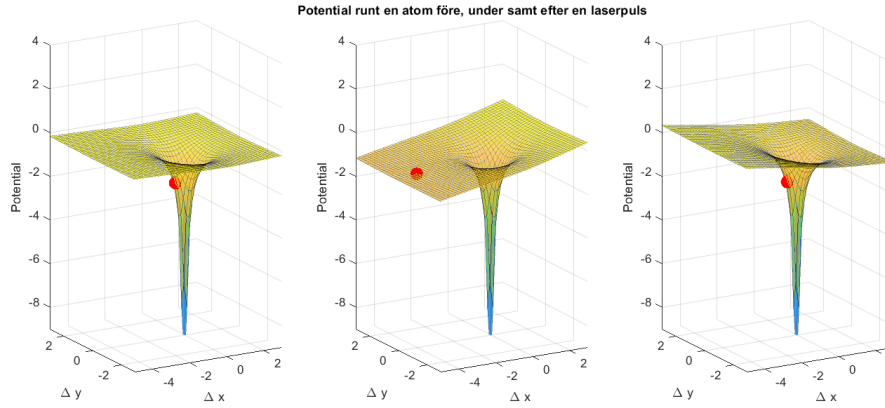
Innan pulsen når en given gasatom, kommer atomens elektroner att vara bundna till atomkärnan i Coulombpotentialen med formen

$$V_c(r) = \frac{a_1}{r}, \quad (2)$$

för en konstant  $a_1$  och där  $r$  är avståndet från kärnan. Potentialen modifieras när pulsen når atomen och kan då modelleras enligt

$$V_c(r, t) = \frac{a_1}{r} + a_2 x \cos(\omega_0 t), \quad (3)$$

där  $a_2$  är amplituden på pulsen och  $x$  är sträckan från kärnan i x-led. Modellen förutsätter att pulsen är tillräckligt bred för att hålla flera våglängder [5]. När pulsen når kärnan kommer därmed den elektriska potentialen att sänkas på ena sidan av atomen. Detta medför att det finns en relativt stor sannolikhet att en elektron tunnlar ut, förflyttar sig genom en potentialbarriär, från Coulombpotentialen och atomen joniseras. Efter joniseringen accelereras elektronen bort från jonen av pulsen, tills det att pulsens elektriska fält byter tecken varefter elektronen får en hastighet i riktning mot jonen. Elektronen kan slutligen bindas till jonen vilket frigör energi motsvarande både bindingsenergin men även den kinetiska energi elektronen byggt upp efter accelerationen av laserpulsen. Energin skickas ut som en foton med en frekvens motsvarande en heltalsmultipel av en frekvens i femtosekundspulsen [5], [2].



Figur 4: Schematisk figur över hur attosekundspulser genereras när femtosekundspulser belyser en ädelgasatom. Den röda pricken markerar en intressant position där en elektron har en hög sannolikhet att finnas sig i respektive bild. Atomkärnan befinner sig i  $[x = 0, y = 0]$ .

Attopulsgenereringen illustreras i figur 4 där potentialen för elektronen runt en atomkärna placerad i origo visas mot avståndet från kärnan. Från vänster till höger visas det hur en elektron är bunden till kärnan enligt potentialen beskriven i ekvation (2), tunnlar ut från kärnan, enligt potential i ekvation (3) med  $\cos(\omega_0 t) = 1$ , och åter binds till kärnan i samma potential då  $\cos(\omega_0 t) = -1$ . Nämnvärt är att pulsen behöver vara mycket stark för att denna process ska ha en märkbar sannolikhet att ske, eftersom joniseringen sker genom tunnling. I uppställningen visad i figur 1 krävs det därmed att laserstrålen är väl riktad för att träffa en liten tvärsnittsytta av molekyler i generationskammaren, markering 5.

## 2.3 Inverkan av luft på IR- och XUV-pulser

För att kunna utföra experiment krävs vakuum genom hela HHG-uppställningen. Vakuum krävs innan generationskammaren för att undvika dispersion av IR-komponenterna i laserpulsen. Efter generationskammaren krävs vakuum för att få en hög transmission av de skapade XUV-pulserna.

### 2.3.1 Pulsbreddning genom dispersionen för IR i luft

IR-pulser genereras av den markerade lasern i figur 1 och färdas fram till generationskammaren, markering 5, där XUV-pulser skapas. IR-pulsernas egenskaper behöver bevaras längs laserbanan för att detta ska kunna ske, vilket innebär att dispersionen i uppställningen behöver minimeras. För att beräkna dispersionen för IR i luft användes Sellmeierekvationen, som beskriver relationen mellan ett ämnes brytningsindex och en fotonens våglängd, se ekvation (4).

$$n(\lambda) = 1 + \sum_i \frac{B_i}{C_i - \lambda^{-2}}. \quad (4)$$

Koefficienterna  $B_i$  samt  $C_i$  är materialkonstanter och  $\lambda$  är ljusets våglängd i vakuum. För luft används typiskt två termer i summan. Koefficienternas värden, vid temperatur 20°C och under ett atmosfärstryck  $\approx 1000$  mbar, presenteras i tabell 1 [6].

Tabell 1: Koefficientvärden tillhörande Sellmeierekvationen. Samtliga värden är angivna i enheten  $\mu\text{m}^{-2}$  och gäller för luft med trycket 1000 mbar samt temperaturen 20°C.

$B_1$	$B_2$	$C_1$	$C_2$
0.05792105	0.00167917	238.0185	57.362

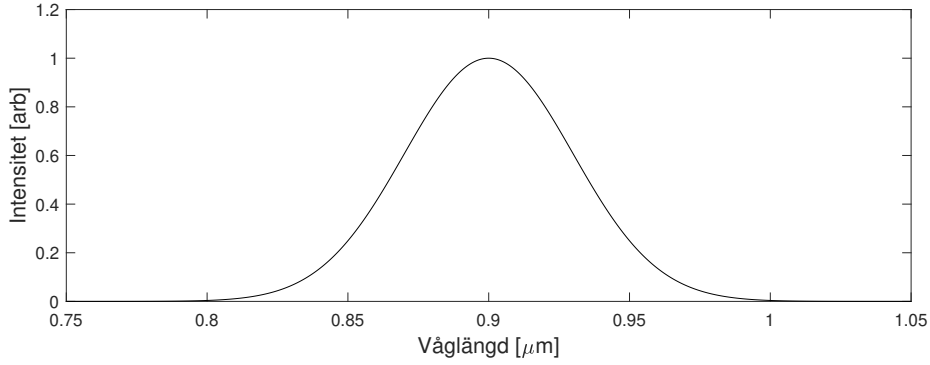
Effekten av brytningsindexets frekvensberoende på en puls kan visas genom att studera en teoretisk puls motsvarande pulsen genererad av lasern i figur 1. Givet att den teoretiska pulsen propagerar i z-led kan det elektriska fältet den ger upphov till tecknas enligt

$$E(z, t) = \sum_{\lambda} E_{\lambda} e^{i\omega(zn(\lambda)/c - t)}. \quad (5)$$

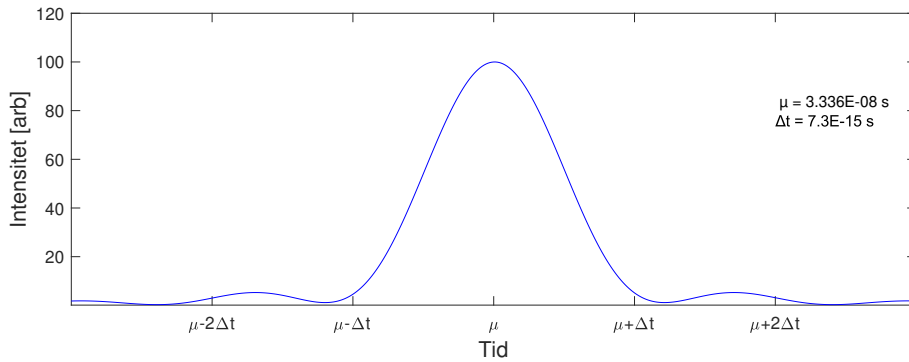
I denna puls betecknar  $z$  avståndet från pulsens generering till mätpunkten och  $\omega = \frac{2\pi c}{\lambda}$  är vinkelfrekvensen.  $E_{\lambda}$  är amplituden på respektive komponents bidrag till det elektriska fältet, och har amplituder fördelade så att intensitetskomponenterna  $I_{\lambda} = E_{\lambda}^2$  är gaussiskt fördelade med väntevärdet  $\lambda_0 = 900$  nm och standardavvikelsen  $\sigma_{\lambda} = 30$  nm.

Amplituden för den ovan beskrivna pulsen har ett maximalt värde  $E_{max} = \sum_{\lambda} E_{\lambda}$ , vilket antas när samtliga termer är i fas. Om pulsen är i vakuum är  $n(\lambda) = 1$  för alla våglängder vilket innebär att det alltid finns en tid så att  $i\omega(zn(\lambda)/c - t) = 0$  för samtliga våglängder. Pulsen kommer därmed att ha samma utseende oberoende av  $z$ -värdet, ett utseende vilket visas i figur 5b. Då  $n(\lambda)$  vid propagering i luft följer ekvation (4) förskjuts faser för de olika vågkomponenterna relativt varandra, vilket medför att pulsens maximala värde är avsevärt lägre samt att pulsens intensitet avtar betydligt långsammare.

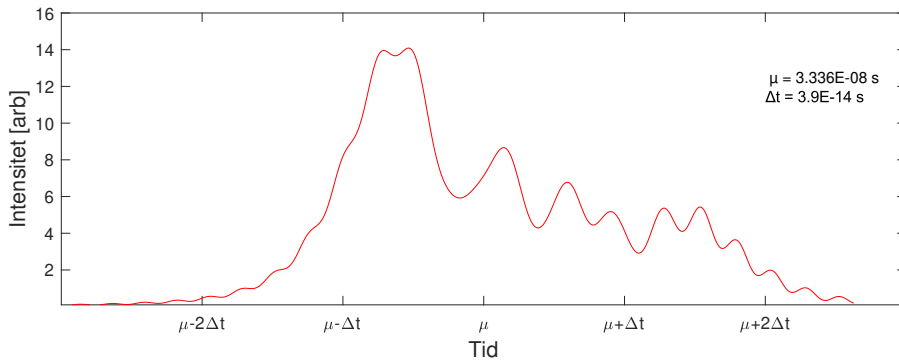
Pulsbredden  $\Delta t$  är definierad som standardavvikelsen för pulsens intensitet i tid, den är 7.3 fs för den ovan beskrivna pulsen vid  $z = 10$  m och propagering genom vakuum. Vid  $z = 10$  m och propagering genom luft blir  $\Delta t$  39 fs. Utbredningen av pulsens intensitetsfördelning över ett större tidsintervall minskar sannolikheten för att pulsen ska kunna jonisera atomer, vilket är nödvändigt för HHG. Med anledning av detta är det önskvärt att minska lufttrycket längs laserbanan där femtopulsen färdas, se figur 1.



(a) Spektrum för testpulsen, intensiteten är gaussiskt fördelad över våglängderna.



(b) Pulsens utseende efter att ha propagerat 10 m i vakuum.



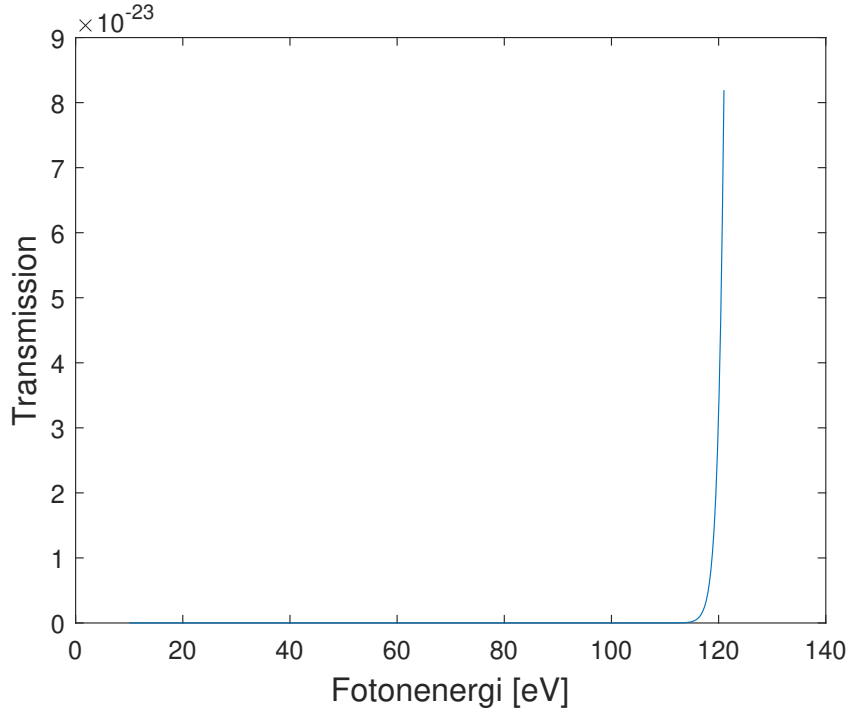
(c) Pulsens utseende efter att ha propagerat 10 m i luft.

Figur 5: Testpulsens spektrum och utseende efter att ha propagerat i luft och vakuum. Notera att pulsbredden är betydligt större och att den maximala intensiteten är betydligt lägre för pulsen som propagerat i luft.  $\mu$  anger intensiteten för den del av pulsen som har färdats exakt 10 m.

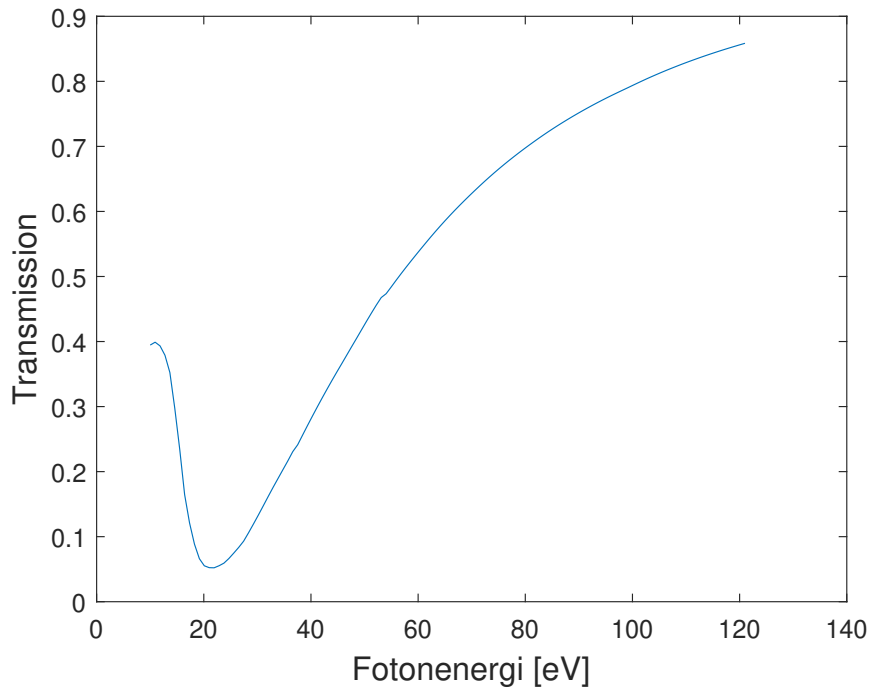
### 2.3.2 Transmission av XUV i luft

I luft med högt tryck är transmissionen av XUV låg. I figur 6 visas transmissionen av XUV i luft, från lågvakuum (LV) till högvakuum (HV), observera att det är stor skillnad i transmission vid jämförelse mellan de tre graferna. Vid trycket 1 mbar blir transmissionen försumbar (figur 6a), men vid början av HV-intervallet ( $10^{-3}$  mbar) börjar transmissionen bli hög (figur 6b). Vid detta tryck är variationen på transmissionen stor och är därför inte

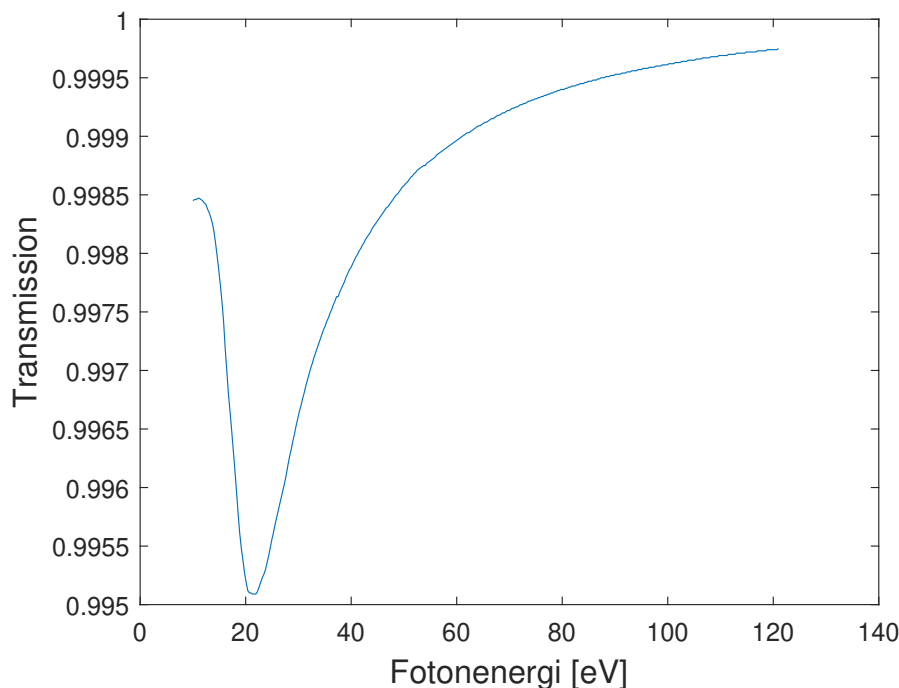
användbar. Det krävs ett tryck på den övre gränsen av HV-intervallet för att absorptionen ska bli näst intill försumbar, se figur 6c. Detta är anledningen till att kamrarna där XUV-strålen färdas måste vara i vakuum, då XUV-strålen kräver högvakuum för att få högsta möjliga transmission. Data för graferna i figur 6 är hämtad från [7].



(a) XUV i luft,  $p = 1$  mbar.



(b) XUV i luft,  $p = 4 \times 10^{-3}$  mbar.



(c) XUV i vakuum,  $p = 7 \times 10^{-6}$  mbar.

Figur 6: XUV under olika tryck i luft. Data är hämtad från [7].

## 2.4 Syfte och mål för arbetet

Syftet med projektet är att utveckla mjukvara, ett LabVIEW-program, för att kontrollera styrsystem för den HHG-uppställning som i nuläget utvecklas på Göteborgs universitet. Med anledning av ovan beskriven teori ska styrsystemet reglera trycket i de kammare lasern leds genom, samt även kontrollera rörelsen för de speglar som ska leda femtosekundslaserpulser genom uppställningen, figur 1.

Styrsystemet är planerat till att kontrollera fyra delar. De två första delarna består av vakuumpumpar och tryckmätare. Vakuumpumparna, rotor-pumpar och turbomolekylär-pumpar (TMP), används till att skapa ett vakuum i kamrarna där lasern ska passera. Tryckmätarna, piranimätare och kallkatodsjonisationsmätare (CCIG, *Cold Cathode Ionization Gauge*), används till att mäta trycket i kamrarna. Den tredje delen består av kameror som används till att bestämma laserns position på speglarna. Den fjärde delen består av piezomotorer. Motorerna används till att möjliggöra förflyttning av speglarna i vakuum med nanometersprecision.

Målet med LabVIEW-programmet som skapas är att det ska vara enkelt att lägga till fler pumpar, mätare, kameror och piezomotorer i den slutgiltiga HHG-uppställningen. Styrsystemet utvecklas i programmeringsspråket LabVIEW då frontpanelen av ett färdigutvecklat LabVIEW-program inte kräver förkunskaper inom programmering för att användas. LabVIEW stöds även av flera hårdvaruutvecklare och är därför lätt att applicera på deras instrument. Genom att i realtid presentera relevant information på frontpanelen med olika grafiska användarsnitt minimeras dessutom behovet för användaren att interagera direkt

med programkoden. Vidare ämnas det att underlätta felsökning genom att spara data och eventuella felmeddelanden.

### 3 Hårdvara

I detta kapitel beskrivs den hårdvara som har använts i detta projekt och kommunikationen mellan hårdvaran och LabVIEW. De specifika modeller som använts listas i tabell 2.

Tabell 2: Lista på hårdvaror som använts under projektet.

Instrument	Tillverkare	Produkt
Rotorpump	Kashiyama Industries	NeoDry 15E-2
Kontrollenhet för turbopump	Leybold	Turbo Drive 400
Turbopump	Leybold	SL-80
Kontrollenhet för tryckmätare	Pfeiffer Vacuum	TPG 256 A
Piranimätare	Pfeiffer Vacuum	TPR 280
CCIG	Pfeiffer Vacuum	IKR 251
Kamera	Flir	Blackfly S GigE

#### 3.1 Vakuumpumpar

Det finns olika vakuumpumpar som kan användas för att skapa ett ultrahögt vakuum (UHV). För detta projekt används det en kombination av två typer av pumpar. Anledningen till att använda två olika typer av pumpar beror på vakuumpumparnas funktionalitet inom de olika tryckintervallen, se tabell 3 för tryckintervall.

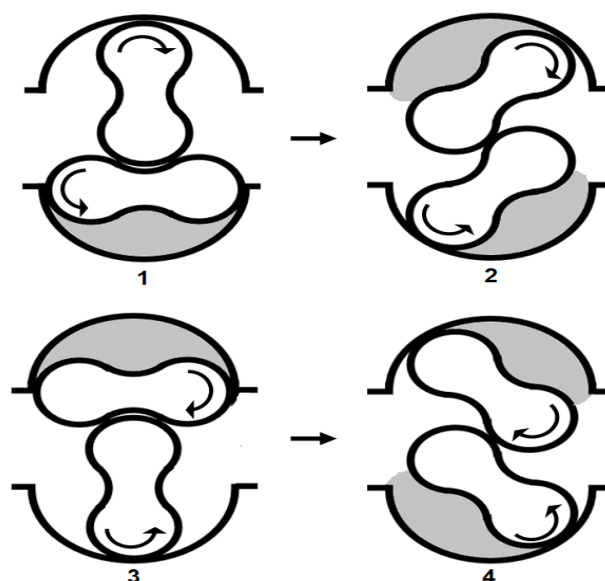
Tabell 3: Vakuumentyper och korresponderande tryckintervall [8].

Vakuumentyp	Intervall
LV	$10^3$ mbar till $10^{-2}$ mbar
HV	$10^{-3}$ mbar till $10^{-7}$ mbar
UHV	$10^{-7}$ mbar till $10^{-16}$ mbar

##### 3.1.1 Rotorpumpar

Rotorpumpar är en typ av grovpump och används där stora mängder gas evakueras under en kortare tid. De används som primära pumpar vid intermediära vakuum för grovpumpning tills att den uppnår tryck där HV-pumpar och UHV-pumpar kan arbeta mer effektivt. Rotorpumpar fungerar genom att två kompressorhjul roterar och växelvis fångar luft mellan kompressorhjulet och höljet för att sedan trycka ut gasen [8], se figur 7.





Figur 7: Princip för rotorpump. När de två kompressorhjulena roterar fångas luft från vänster i de markerade grå områdena, för att sedan pumpas ut åt höger i figuren.

Rotorpumparna överför stora mängder gas under en kortare tid vilket gör dem effektiva som initialpumpar, men ett problem med rotorpumpar är baksug. Eftersom rotorpumpar saknar ventiler är de beroende av närgång mellan kompressorhjulena och höljet för att minimera baksug. Detta är en av anledningarna till att rotorpumpar inte är effektiva för HV och UHV [8].

### 3.1.2 Turbopumpar

En turbopump eller en turbomolekylärpump, är i princip en turbin som komprimerar luft genom momentumöverföring från de roterande bladen till gasen [8]. Kommersiella turbopumpar kan komma ner till tryck som ligger mellan  $10^{-9}$  mbar och  $10^{-10}$  mbar och används därmed för generering av HV och UHV. Pumpar som TMP och andra UHV-pumpar tenderar att inte fungera under höga tryck (atmosfärstryck). Anledningen till detta är att under laminärt flöde ( $>10^{-3}$  mbar) förhindras turbinens arbete av de höga antalet kollisioner mellan partiklarna i gasen. Detta innebär att pumpar som UHV-pumpar har svårigheter med att pumpa ut stora mängder materia och utrustningen tenderar att skadas vid felanvändning. Därmed måste de användas i kombination med en rotorpump eller annan grovpump [8].

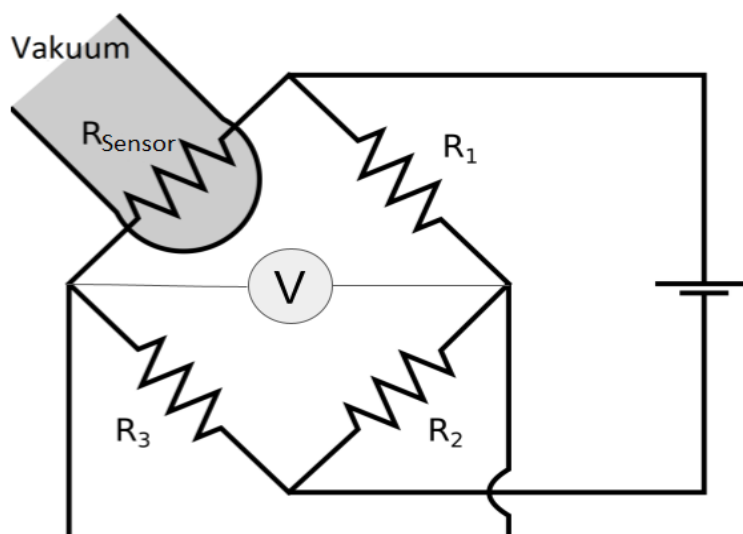
## 3.2 Tryckmätare

I detta arbete användes en piranimätare och en kallkatodjonisationsmätare för att mäta tryck. Båda är indirekta mätare som bestämmer trycket genom att mäta gasens termiska konduktivitet. När trycket sjunker minskar gasens densitet vilket medför lägre konduktivet [8].

### 3.2.1 Piranimätare

Vid tryckmätning i en piranimätare skickas ström genom en volframtråd. Tråden, som är omgiven av den gas som ska mätas, genererar värme. När en gasmolekyl kolliderar med tråden överförs värme till gasmolekylen. Värmeförlusten är proportionell mot gasens tryck och vid lågt tryck blir den fria medelväglängden lång. Detta leder till att den termiska konduktiviteten minskar då den är proportionell mot gasens densitet och invers proportionell mot den fria medelväglängden [8].

Piranimätaren använder en Wheatstonebrygga, se figur 8, för att bestämma resistansen i en okänd resistor. Metoden kan detektera små variationer av resistansen. Detta gör att den okända resistorn kan användas som en sensor. I en piranimätare utgörs sensorn av en volframtråd som beskrivs ovan. Tråden hålls på en konstant temperatur vilket gör att om gasens värmekapacitet minskar krävs det mindre energi för att hålla värmen konstant i tråden. Voltmätaren i Wheatstonebryggan mäter spänningen i tråden som sedan konverteras till tryck [9]. Den piranimätare som användes kunde mäta tryck från atmosfärstryck ner till  $5 \times 10^{-3}$  mbar.

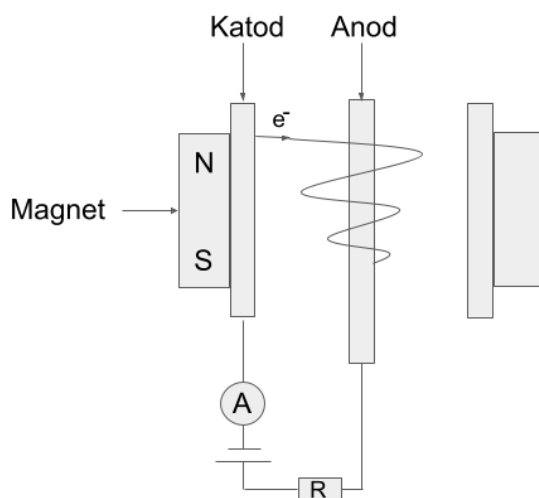


Figur 8: Piranimätarens uppbyggnad. En av resistorerna i Wheatstonebrygga är en volframtråd som hålls på en konstant temperatur och är omgiven av den gas som ska mätas. Om gasens värmekapacitet minskar krävs det mindre energi för att hålla temperaturen konstant i tråden. Den uppmätta spänningen i voltmätaren konverteras sedan till tryck [9].

### 3.2.2 Kallkatodjonisationsmätare

En kallkatodjonisationsmätare består av en anodstav, i vårt fall av molybden, som är omgiven av en cylindrisk kammare med den gas som ska mätas. Kammaren består av ett

katodhölje. Anoden och katoden är sammankopplade med en amperemeter, en spänningskälla och en resistor. En magnet omsluter katodhöljet och skapar ett magnetfält inuti kammaren, se figur 9. Katoden avger elektroner med fältemission i riktning mot anoden [10]. Magnetfältet ger upphov till att elektronerna färdas i en spiralformad bana vilket gör att elektronernas väg till anoden blir längre. Detta leder till att sannolikheten för att en elektron kolliderar med en gasmolekyl ökar och vid kollision joniseras molekylen. Den positiva gasjonen kommer sedan att röra sig mot katoden där strömmen som jonen avger läses av amperemetern, som är kopplad till katoden, och omvandlas till tryck [8]. Den CCIG som användes kunde mäta tryck från  $2 \times 10^{-9}$  mbar till  $10^{-2}$  mbar.



Figur 9: Kallkatodjonisationsmätarens uppbyggnad. En katodhöljes kammare är fylld av den gas som ska mätas. Katoden avger elektroner som påverkas av magnetens magnetfält, vilket gör att elektronerna färdas i helixbanor mot anoden. Den nya elektronrörelsen gör att sannolikheten för en kollision med en gasmolekyl ökar. Gasjonens ström tas upp av katoden och mäts av amperemetern som är kopplad till katoden och anoden [8].

### 3.3 Piezomotorer

Piezomotorers funktion är att förflytta (translatera) eller rotera ett objekt. En piezomotor fungerar genom att man skickar en ström genom en piezokristall som sedan antingen förlängs, böjs eller båda. Varje enskild kristalls förändring är väldefinierad och utvecklad utefter det behov som finns i apparaturen. När en elektrisk puls går igenom kristallen deformeras den, när pulsen passerat, relaxerar kristallen och återgår till sitt ordinarie tillstånd. På detta sätt kan den arbeta under höga frekvenser för att snabbt utföra förflyttningar med stor noggrannhet [11]. En viktig anledning till att just använda piezomotorer i vakuumsystem är att de inte har något behov av olja för smörjning. Behov av smörjning är ett problem för vakuumsystem då det innebär risker för kontaminering.

### 3.4 Hårdvarukommunikation

Nedan följer en överblick över hur data skickas mellan de olika instrumenten, samt vilken data som skickas.

#### 3.4.1 Kommunikation med tryckmätare och turbopumpar

Kommunikationen med tryckmätare och turbopumpar utförs genom kommunikationsstandarden RS-232. Informationen skickas i standarden som ett antal logiska bitar, där spänningar mellan 5 V och 25 V representerar en logisk nolla och spänningar mellan −5 V och −25 V representerar en logisk etta. Hastigheten med vilken bitarna skickas anges i baud, där en baud är ekvivalent med en bit per sekund [12]. Datan skickas i paket där varje paket inleds med en startbit, med värdet noll, följt av ett antal databitar. Efter databitarna skickas, vid efterfrågan, en kontrollbit och paketet avslutas med en stoppbit vilken har värdet ett [13]. Parametervärden för kommunikation med kontrollenheten för tryckmätarna samt för kommunikation med turbopumpen anges i tabell 4.

Tabell 4: Parametervärden för kommunikation med kontrollenhet för tryckmätare, TPG 256 A, samt turbopumpkontrollen TD400. En jämn kontrollbit antar värdet ett om det totala antalet bitar med värdet ett i paketet är jämnt.

Instrument	baud	Antal databitar	Kontrollbit
TPG 256 A	9600	8	Nej
TD400	19200	8	Jämn

I LabVIEW representeras datan som skickas och tas emot av instrumentet, som en textsträng. Varje symbol i textsträngen motsvarar det numeriska värdet som databitarna i motsvarande datapaket (byte) kodar för binärt. Ett exempel på hur informationen *Hello world!* kan representeras som en sträng, decimala samt binära värden visas i tabell 5. Decimalrepresentationen för samtliga textsymboler som används i LabVIEW återfinns i appendix A.

Tabell 5: Olika representationer av *Hello world!*

Datapaket	1	2	3	4	5	6
Text	H	e	l	l	o	<i>mellanslag</i>
Decimalt värde	72	101	108	108	111	32
Binärt värde	01001000	01100101	01101100	01101100	01101111	00100000
Datapaket	7	8	9	10	11	12
Text	w	o	r	l	d	!
Decimalt värde	119	111	114	108	100	33
Binärt värde	01110111	01101111	01110010	01101100	01100100	00100001

### 3.4.2 Kommunikation med kameror

Information mellan kamera och dator skickas via ethernetanslutning. När en bild skickas från kameran Blackfly S GigE, specificeras ett pixelvärde mellan 0 och 255 för varje enskild pixel, där 0 representerar en svart pixel och 255 en maximalt ljus pixel. Utöver att skicka bildinformation kan kameramodellen även ta emot information som specificerar kameraattribut såsom exponeringstid och dylikt.

## 4 Metoder och kommunikationsprotokoll

För att hämta data från och styra instrumenten användes olika LabVIEW-funktioner. I detta avsnitt presenteras implementeringen av dessa.

### 4.1 Tryckreglering

Inledningsvis skapades sub:VI till alla grundläggande funktioner. Funktionerna skulle kommunicera med, läsa av samt logga data från pumpar och mätare. Vidare angav funktionerna datorns interna tid, samt hämtade och loggade datorns tid. De skulle även rita en graf av datan mot tiden i realtid samt skriva ut felmeddelanden om ett fel uppstått i kommunikationen med pumparna. Detta för att underlätta en eventuell felsökning av utrustningen. Med alla sub:VIs skapades ett program där de samverkade och initierade varandra. Efter vi hade uppnått automatisering av programmet kunde vi lägga in funktioner som gjorde programmet mer användarvänligt. I figur 10 visas en schematisk bild över testuppställningen vi använde för att testa vårt program.



Figur 10: Schematisk bild av testuppställningen. Rotorpumpen är igång hela tiden under vakuumgenereringen. Piranimätaren är inkopplad mellan rotorpumpen och TMP för att endast mäta trycket vid rotorpumpen. Kalkatodsjonisationsmätaren är kopplad direkt till testkammaren och mäter trycket från HV till UHV, vilket uppnås genom att använda TMP.

#### 4.1.1 Kommunikation med tryckmätare

För att erhålla tryckvärden från piranimätare samt CCIG kommunicerar LabVIEW med kontrollenheten TPG 256 A, genom funktionerna listade i tabell 6. Dessa funktioner är en del av standardbiblioteket i LabVIEW.

Tabell 6: Funktioner i LabVIEW som användes för kommunikation med mätinstrument.

Funktioner	Förklaring
VISA configure serial port	Öppnar kommunikation med instrumentet genom angiven COM-port.
VISA write	Skriver information till instrumentet
VISA read	Läser information instrumentet skickat
VISA close	Stänger kommunikation genom angiven COM-port

*VISA configure serial port* kräver att parametrar angivna i tabell 4 anges, *VISA write* tar en textsträng som parameter och *VISA read* ger en sträng som svar. Pseudokod för att hämta och läsa av ett tryckvärde presenteras nedan.

```

1  function getPressure{
2      VISA configure serial port(baud = 9600,
3      databitar per paket = 8, ingen kontrollbit, port = COM1);
4
5      while(mätning pågår){
6          VISA write("PR1", COM1); %PR1 för sensor 1
7          vänta på att 3 bytes har skickats;
8          enq = text-representation av decimalt värde 5;
9          VISA write(enq, COM1);
10         vänta på att 15 bytes har skickats;
11         dataString = VISA read(COM1);
12         felkod = dataString(3);
13         tryck = string2number(dataString(5:end-2));
14         if(felkod != 0){
15             return(error); %Rapporterat error styrs av felkod
16         }
17         return(tryck)
18     }
19     VISA close(COM1)
20 }
```

I koden konfigureras COM-porten varefter tryckvärden hämtas och returneras kontinuerligt. Notera att *VISA write* behöver kallas två gånger. Från rad ett till fem initieras samt konfigureras kommunikationen med tryckmätarna. På rad sex skickas PR1, pressure sensor 1, för att sensorn ska utföra och spara data från en tryckmätning. Som svar på detta kommando skickas textrepresentationen av [6, 13, 10] = [Positiv bekräftelse, Enter, ny

rad], som positiv bekräftelse på att kommandot tagits mot, alternativt [21, 13, 10] = [negativ bekräftelse, Enter, ny rad] som negativ bekräftelse. Då det oavsett skickas tre bytes väntar programmet på att det finns tre bytes att läsa innan nästa kommando skickas till tryckmätaren. *VISA write* på rad 9 skickar textrepresentationen av 5, ett standardiserat textkommando för att information ska skickas. Som svar skickas ytterligare 12 bytes från turbopumpen och därmed finns totalt 15 bytes vid detta tillfälle att läsa. Datan läses av på rad 11 och har strukturen [6, 13, 10, "0,6.855E-3", 13, 10] där index 0 till 2 finns kvar från förra kommandot. 0,6.855E-3 innehåller en statusbyte, "0", följt av tryckvärdet vilket i detta fall har värdet  $6.855 \times 10^{-3}$  mbar. Att statusbyten har värdet noll visar att mätningen har lyckats, övriga värden på statusbyten indikerar ett misslyckande.

Noterbart för tryckmätarna är att den numeriska tryckdatan skickas som strängrepresentationer. 6.855E-3 skickas som åtta byte, det vill säga en byte för varje symbol, istället för att skickas som ett binärt värde.

#### 4.1.2 Kommunikation med turbopumpar

Mjukvarukommunikationen med pumpkontrollen TD400 är snarlik kommunikationen med tryckmätarna då samma funktioner, listade i tabell 6, används. Den största skillnaden är att data skickas till och från pumpen som en sträng vilken alltid består av 24 tecken, samt att numeriska värden inte skrivs ut i klartext utan skickas som de tecken vilka har numeriska representationer motsvarande de i appendix A. Numeriska värden som är för stora för att representeras av ett tecken, en byte, skickas genom att två tecken, 16 bits, tolkas som ett gemensamt binärt värde  $\in [2^0, 2^{16} - 1]$ . Algebraiskt är ihopslagningen ekvivalent med  $2^8 \times \text{Byte } n + \text{Byte } m$ . Pseudokoden för pumpkommunikationen följer nedan.

```

1  function TD400Control(boolean ON, int req){
2      % ON = true om pumpen ska vara aktiv
3      VISA configure serial port(baud = 19200,
4          databitar per paket = 8, Jämn kontrollbit, port = COM2);
5
6      pumpCmd = zeros(24,1); % array av numeriska nollor
7      pumpCmd(0) = 2; % textkommando, start av data
8      pumpCmd(1) = 22; % Storlek på data = 24-(start och kontroll);
9      pumpCmd(3) = 16; % Data ska skickas från pumpen
10     pumpCmd(4) = req; % Vilken data som ska skickas
11     pumpCmd(11) = 4; %Pump ska kontrolleras via LabVIEW
12
13     while(mätning pågår){
14         pumpCmd(12) = boolean2number(ON); %Pump av eller på
15         pumpCmd(23) = iterativ XOR över pumpCmd 0 till 22,
16         VISA write(array2string(pumpCmd), COM2);
17         vänta på att 24 bytes har skickats;
18         Data = string2numberarray(VISA read(COM2));
19         reqData = Merge(Data(9),Data(10)); %Merge enligt text ovan
20         Frekvens = Merge(Data(13),Data(14));
21     }
22 }
```

```

22      Temperatur = Merge(Data(15),Data(16));
23      Spänning = Merge(Data(21),Data(22));
24      Errorkod = Merge(Data(11),Data(12)); % 0 om inget error
25      Return(reqData,Frekvens,Temperatur,Spänning,Errorkod)
29  }
30 }

```

Liksom för tryckprogrammet initialiseras först COM-porten varefter data skickas och hämtas kontinuerligt i turbopumpen. Pumpkommandot består alltid av 24 bytes, där index 0, 1, 3, 4, 11, 12 och 23 håller information som behövs för den typ av kommunikation som utförs. Samtliga index utöver 12 och 23 hålls konstanta över hela mätningen. Värdet på index 12 sätts till 1 om turbopumpen ska vara på och till 0 om den ska vara av. På rad 16 beskrivs kontrollbiten med index 23, vilken beräknas med hjälp av en XOR, exclusive or,  $XOR(A,B) = 1$  om exakt en av A och B har värdet 1. I ett första steg utförs XOR på bitarna med samma position i pumpcmd(0) och pumpcmd(1),

$$XOR(pumpcmd(0),pumpcmd(1)) = XOR(00000010,00010110) = 00010100.$$

Operationen upprepas sedan iterativt med den resulterande bitsekvensen och nästa index i pumpcmd upp till och med index 22.

## 4.2 Optik

Genom att kommunicera med en kamera placerad bakom en spegel, kan vi få information om laserstrålens position och intensitet. Med denna data kan vi beräkna laserstrålens masscentrum i kamerabilden. Detta är nödvändigt för att senare kunna förflytta spegeln med hjälp av en piezomotor.

Kommunikationen med kameran sker med hjälp av funktioner i LabVIEW som är listade i tabell 7. Dessa funktioner ges av mjukvaran *Vision Acquisition Software* från *National Instruments*. Bilderna hämtas med 8 bitar per pixel och ritas ut grafiskt i realtid.



Tabell 7: *Vision Acquisition Software* funktioner vilka används i LabVIEW för kommunikation med kameran.

Funktioner	Förklaring
IMAQdx Open Camera	Öppnar kommunikation med en kamera genom USB-anslutning.
IMAQdx Configure Acquisition	Anger om ackvisitionen är kontinuerlig eller om det endast är en bild.
IMAQdx Start Acquisition	Startar ackvisitionen som tidigare konfigurerats.
IMAQ Create	Skapar ett tillfälligt minne för bilden.
IMAQdx Get Image2	Hämtar den konfigurerade bilden.
IMAQdx Stop Acquisition	Stoppar ackvisitionen av bilder.
IMAQdx Close Camera	Stoppar och avslutar kommunikationen med den kameran som tidigare öppnats.
IMAQ Centroid	Beräknar bildens masscentrum för x- respektive y-koordinat.
IMAQ Threshold	Ändrar samtliga pixelvärden inom ett intervall till ett specificerat värde. Övriga pixelvärden ändras till noll.
IMAQ Overlay Oval	Ritar en oval på en bild.

#### 4.2.1 Struktur för att hämta bildinfo via LabVIEW

Informationen i bilder kan hämtas via den struktur som presenteras i pseudokoden nedan. 'cam0' representerar här en referens till den kamera som behandlas. På rad 2 till 4 konfigureras kameran och en kontinuerlig ackvisition påbörjas, varefter bilder kontinuerligt returneras på rad 5 till 7. Avslutningsvis stängs ackvisitionen och kommunikationen med kameran av på rad 8 och 9.

```

1  function getImageInfo(){
2      IMAQdx Open Camera(cam0);
3      IMAQdx Configure Acquisition(Kontinuerligt hämtande av bilder);
4      IMAQdx Start Acquisition(cam0);
5      While(mätning pågår){
6          return(IMAQdx Get Image2(cam0));
7      }
8      IMAQdx STOP Acquisition(cam0);
9      IMAQdx Close Camera(cam0);
10 }
```

#### 4.2.2 Beräkning av laserns position i en bild

Beräkningen av laserns position i en bild utförs genom att först sätta alla pixelvärden under ett tröskelvärde till noll och alla värden över tröskelvärdet till 255. Detta steg

genomförs med *IMAQ Threshold* och utförs för att minimera störningar av bakgrundsljus. Efter manipulationen används *IMAQ Centroid* i syfte att beräkna koordinaterna  $[\hat{x}, \hat{y}]$  för intensitetens masscentrum. Funktionen beräknar masscentrum genom

$$\begin{aligned}\hat{x} &= \frac{1}{K} \sum_{x,y} xp(x,y), \\ \hat{y} &= \frac{1}{K} \sum_{x,y} yp(x,y), \\ K &= \sum_{x,y} p(x,y)\end{aligned}\tag{6}$$

där  $p(x, y)$  är pixelvärdet i pixeln med koordinaterna  $[x, y]$ . Origo placeras i bildens övre vänstra hörn. Masscentrum antas sammanfalla med laserns position. Slutligen markeras masscentrum med en cirkel. Pseudokoden för den beskrivna bildmanipulationen följer nedan:

```
1  function findCM(image, thFilter, thValue){
2      if(thFilter == true){
3          image = IMAQ Threshold(image, thValue, 255);
4      }
5      cm = IMAQ Centroid(image);

6      image =
          IMAQ Overlay Oval(image, radie = 100 pixlar, mittpunkt = cm);
7      return(image);
8  }
```

I koden motsvarar *thFilter* ett booleskt värde vilket avgör om den ovan beskrivna tröskelvärdesmanipulationen ska utföras, och *thValue* motsvarar tröskelvärdet som avgör vilka pixelvärden som ska sättas till 0 respektive 255. Efter manipulationen med hjälp av tröskelvärde på rad 2 till 4 följer den ovan beskrivna markeringen av bildens masscentrum på rad 5 och 6, varefter bilden returneras på rad 7.

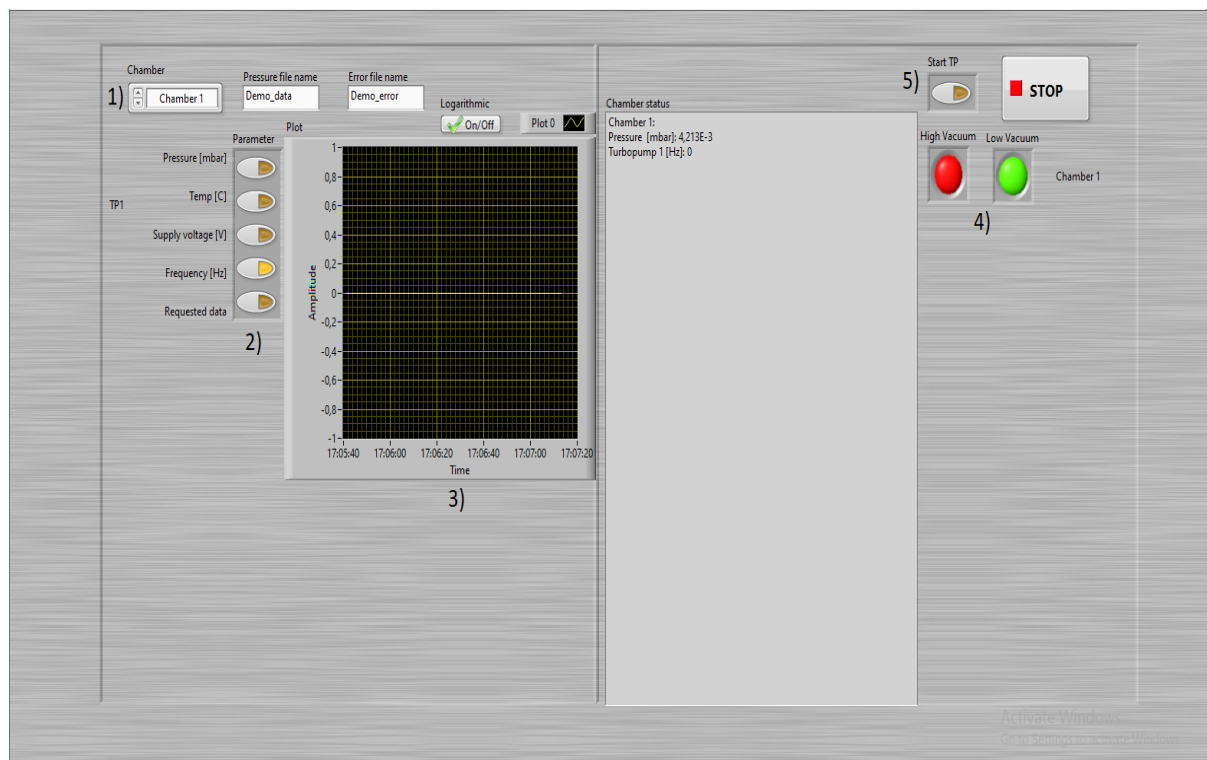
## 5 Resultat & Diskussion

I detta avsnitt kommer LabVIEW-programmen samt resultaten av programmen presenteras och diskuteras. Avsnittet är indelat för de två funktionerna, tryckreglering och optik. För tryckregleringen och optiken presenteras strukturen på frontpanelen och namn samt förklaringar till de sub:VIs vi skapat. Tryckregleringen presenterar även struktur på loggning av data och felmeddelanden.

Till följd av flera veckors försenade leveranser av utrustning saktades arbetsprocessen ner, vilket medförde att mindre tid än planerat kunde ägnas åt optikdelen av styrsystemet och därmed uteblev sammankopplingen av piezomotorn och kameran.

## 5.1 Tryckreglering

Slutprodukten av LabVIEW-programmet för tryckreglering presenteras av programmets frontpanel som visas i figur 11.

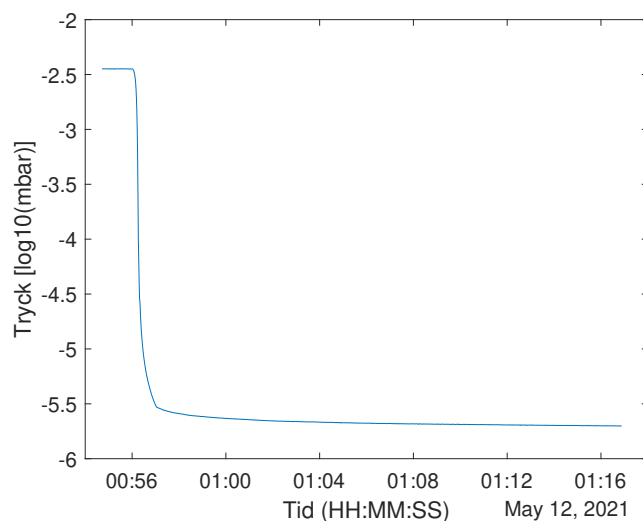


Figur 11: Frontpanel för tryckregleringsprogrammet. 1) Val av kammare. 2) Val av parameter som ska ritas grafiskt mot tiden. 3) Graf som ritas en vald parameter mot tiden. 4) Signaler för när det är högt och lågt vakuum. 5) Start och stopp för turbopump.

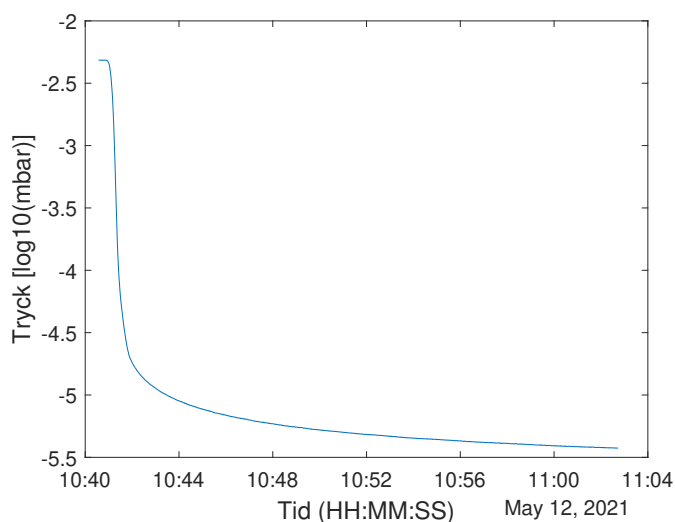
Styrsystemet för vakuumpumparna samt vakuummätarna består huvudsakligen av två while-loopar. Den yttre while-loopen används för att kunna skapa en ny textfil för att logga data och en textfil för att logga felmeddelanden. Båda while-looparna kan stoppas genom ett manuellt stopp av programmet. Den inre while-loopen används för att efter varje iteration kunna interagera samt hämta och behandla data från tryckmätarna och turbopumpen. Datan som man får fram från tryckmätaren är enbart trycket, men från turbopumpen beror det på vad man specificerat. Standarddata från turbopumpen är frekvens (rotationshastighet), spänning samt temperatur. Ytterligare data som kan begäras från turbopumpen är exempelvis motorströmmen. Utanför den yttre while-loopen används en sub:VI som skapar ett kluster av information som meddelar från vilken COM-port som specifik data ska hämtas, se figur 14.

För att rita en graf används den data som hämtades från systemen, detta ritas utefter tid (visas som timma, minut och sekund (HH:MM:SS)) och uppdateras med varje iteration tillsammans med textfilen där data loggas. Grafen som ritas kan väljas genom att trycka på respektive på- och av-knapp i frontpanelen för respektive parameter. I figur 12 visas tryckregleringen när turbopumpen aktiveras för två testkammare med olika volym och hur trycket förändras med tiden när turbopumpen stängs av. I figur 13 ses strukturen på

de sparade textfilerna.



(a) Grafiskt exempel av tryck i en mindre kammare.



(b) Grafiskt exempel av tryck i en större kammare.

Figur 12: Exempelgrafer på vakuumbildning ritade i Matlab, för en mindre kammare och en större kammare. Notera tidsskillnaden.

En andra textfil skapas också för att spara möjliga felmeddelanden som uppkommer och dessa sparas med tidpunkt för eventuell händelse. Felmeddelanden kommer antingen direkt från hårdvarans färdigprogrammerade felmeddelanden eller från egenprogrammerade felmeddelanden som skapats i samband med automatiseringsprogrammet. En del av de felmeddelanden som finns kan hittas i tabell 8.

Tabell 8: Egendefinierade felmeddelanden som kan uppkomma.

Errorkod	Beskrivning
5001	Timeout i 'waitfordata', Låg timeouttid.
5002	Oväntat antal databytes i 'waitfordata'.
-8001	Felaktigt kommando eller att com-porten inte är kalibrerad.
-8002	Fel vid tryckavläsning. Värden listade i beskrivning till 'getPressure'.
-8003	Turbopumpen(TD400) har inte skickat 24 byte data.
-8004	Error från TD400, beskrivet i medföljande manualen [13].

Date: 2021-04-14

Ch n TP m: turbopump number m in chamber number n

Pressure (Pr) is measured in mbar and Frequency (freq) in Hz

Time	Ch 1 Pr	Ch 2 Pr	Ch 1 TP 1 freq	Ch 2 TP 1 freq	Ch 2 TP 2 freq	Ch 2 TP 3 freq
16:54:43	6,850E-3	3,563E-3	0,000E+0	0,000E+0	0,000E+0	0,000E+0
16:54:44	6,884E-3	3,560E-3	0,000E+0	0,000E+0	0,000E+0	0,000E+0
16:54:45	6,875E-3	3,563E-3	0,000E+0	0,000E+0	0,000E+0	0,000E+0
16:54:46	6,880E-3	3,569E-3	0,000E+0	0,000E+0	0,000E+0	0,000E+0
16:54:47	6,884E-3	3,556E-3	0,000E+0	0,000E+0	0,000E+0	0,000E+0
16:54:48	6,884E-3	3,563E-3	0,000E+0	0,000E+0	0,000E+0	0,000E+0
16:54:49	6,889E-3	3,563E-3	0,000E+0	0,000E+0	0,000E+0	0,000E+0
16:54:50	6,894E-3	3,563E-3	0,000E+0	0,000E+0	0,000E+0	0,000E+0
16:54:51	6,889E-3	3,560E-3	0,000E+0	0,000E+0	0,000E+0	0,000E+0
16:54:52	6,889E-3	3,566E-3	0,000E+0	0,000E+0	0,000E+0	0,000E+0
16:54:53	6,865E-3	3,560E-3	0,000E+0	0,000E+0	0,000E+0	0,000E+0
16:54:54	6,855E-3	3,560E-3	0,000E+0	0,000E+0	0,000E+0	0,000E+0
16:54:55	6,825E-3	3,560E-3	0,000E+0	0,000E+0	0,000E+0	0,000E+0
16:54:56	6,845E-3	3,566E-3	0,000E+0	0,000E+0	0,000E+0	0,000E+0
16:54:57	6,850E-3	3,566E-3	0,000E+0	0,000E+0	0,000E+0	0,000E+0

(a) Struktur av sparad data.

Date: 2021-04-28

Registered error messages for each time

10:41:32 Chamber 1:

-8003 StartAndData\_TD400, Have not recieved (enough) data from turbo pump

10:41:33 Chamber 1:

-8003 StartAndData\_TD400, Have not recieved (enough) data from turbo pump

10:41:34 Chamber 1:

-8003 StartAndData\_TD400, Have not recieved (enough) data from turbo pump

10:41:35 Chamber 1:

-8003 StartAndData\_TD400, Have not recieved (enough) data from turbo pump

10:41:36 Chamber 1:

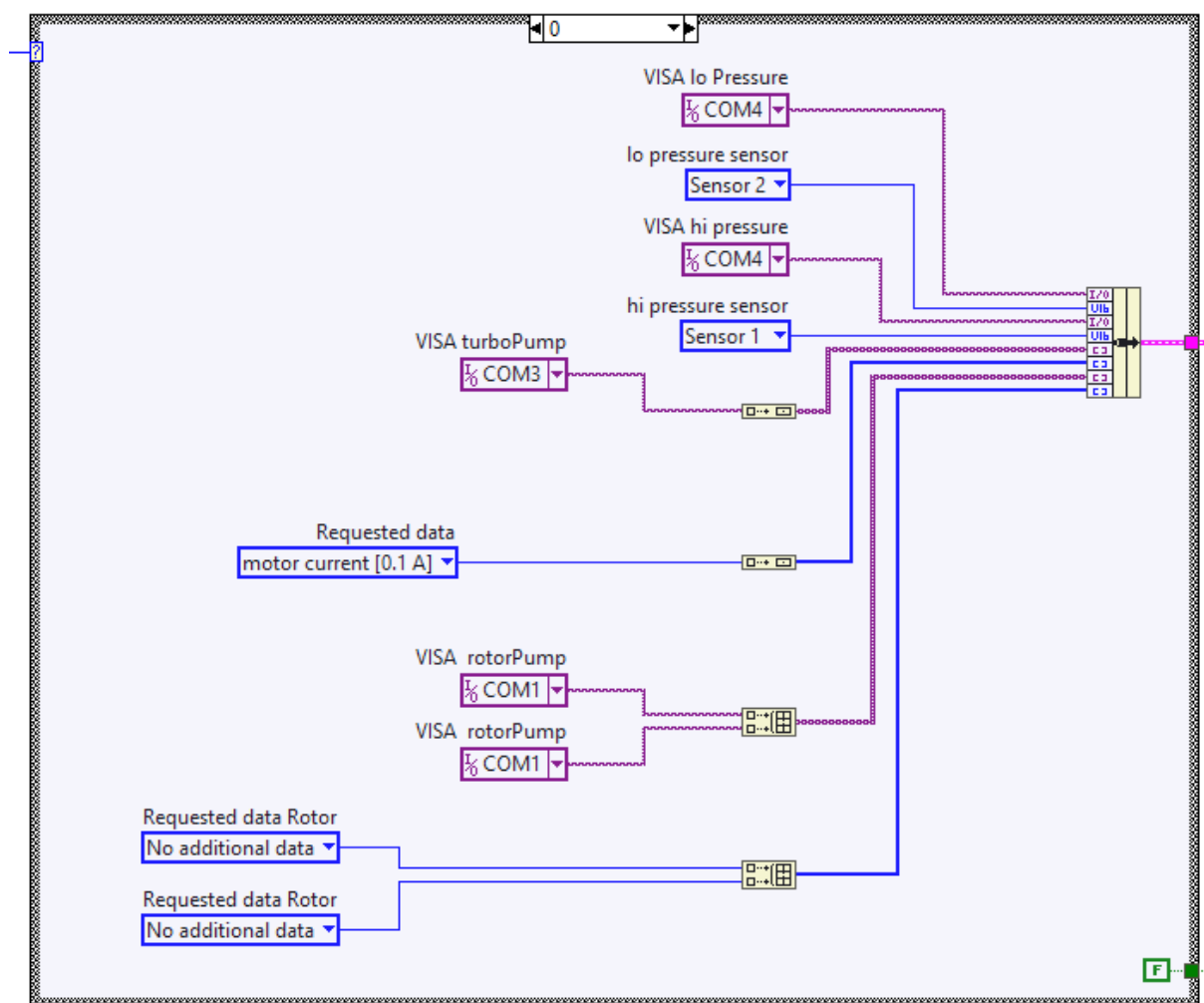
-8003 StartAndData\_TD400, Have not recieved (enough) data from turbo pump

10:41:37 Chamber 1:

-8003 StartAndData\_TD400, Have not recieved (enough) data from turbo pump

(b) Struktur på sparad felmeddelande. Observera att under denna testkörning av programmet var ingen pump inkopplad för att forcera felmeddelanden.

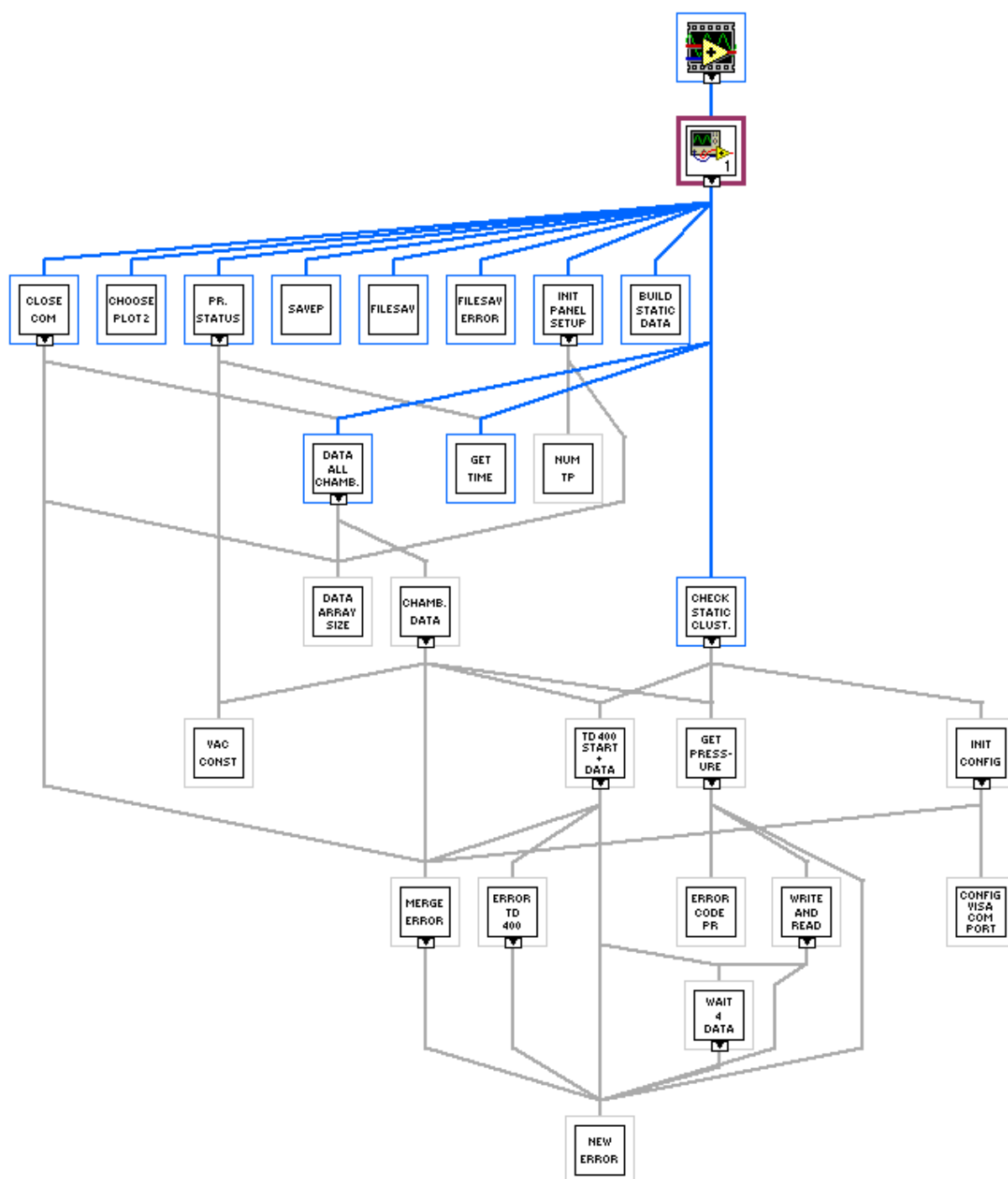
Figur 13: Exempel på sparad data och felmeddelanden.



Figur 14: Informationskluster, informationen består av vilken COM-port som ska användas och även vilken parameter som ska hämtas.

I figur 14 visas innehållet i det informationskluster som håller kontrollinstruktioner för turbopumpar och tryckmätare. Varje sådant kluster håller adresser till de olika instrumenten samt även *requested data*, efterfrågad data, som bestämmer vilken extra data som skickas från turbopumpen utöver standarddata. Varje sådant kluster motsvarar instruktionerna som går till en kammare och samlas i en vektor med övriga kluster. Vektorn skickas sedan till, och kontrollerar funktionen av, huvudprogrammet. Notera att tryckmätarna, markerade som *lo pressure sensor* respektive *hi pressure sensor*, skickas som enskilda värden vilket innebär att enbart en tryckmätare av varje sort, pirani samt CCIG, kan styras från varje kluster. I kontrast till detta skickas efterfrågad data samt adresser till pumpar som vektorer, vilket innebär att ett kluster kan kontrollera fler turbopumpar.

För att implementera styrsystemet för trycket i uppställningen, presenterad i figur 1, krävs det enbart att fler kluster skapas, ett för varje kammare. Vid fall där det installeras fler mätare per kammare behöver dessa kontrolleras av fler kluster. Alla skapade sub:VI från tryckprogrammet och hur de är kopplade finns i figur 15. Namn och förklaringar till sub:VI vi skapat visas i tabell 9. Hierarkin vilken beskriven hur dessa sub:VI kallas av huvudprogrammet visas i figur 15.



Figur 15: Alla skapade sub:VI och hur de är kopplade till huvudprogrammet.

Tabell 9: Namn och förklaringar för sub:VIs vi skapat för tryckprogrammet.

Sub:VIs	Förklaring
Init panel setup	Sätter startvärden för de indikatorer som visas på frontpanelen. Exempelvis för att turbopumparna inte ska starta direkt, se markering 5 i figur 11. Kallas en gång när huvudprogrammet startas.
Build static data	Bygger informationsklustret som visas i figur 14.

PR. status	Läser tryck, frekvenser och felmeddelanden från turbopumpar och tryckmätare för att skapa ett textmeddelande som visas på frontpanelen, markering 3 i figur 11. Kallas varje gång ny data har hämtats från samtliga instrument.
SaveP	Sparar tryck och frekvensdata i ett dokument, se figur 13a. Skriver ny data till dokumentet varje gång samtliga tryck och frekvenser uppmätts.
FileSav	Skapar filen för SaveP att skriva data till, samt skriver datum och övrig inledande information i dokumentet. Kallas en gång när huvudprogrammet startas samt varje gång det blir en ny dag.
FileSav Error	Skapar ett dokument för att rapportera felmeddelanden, se figur 13b. Kallas samtidigt som FileSav.
Close COM	Avslutar kommunikationen med samtliga instrument och stänger av turbopumparna. Kallas när huvudprogrammet avslutas.
Choose plot 2	Väljer vilken parameter som visas i grafen på frontpanelen, figur 11.
NUM TP	Räknar antalet turbopumpar per kluster. Kallas av Init panel setup. Datan används för att bestämma formatet på de textfiler som sparas.
Get Time	Returnerar datorns interna tid som en sträng, HH:MM:SS.
Data all chamb.	Hämtar tryck och annan data från turbopumpar och tryckmätare från samtliga kammare och sparar denna som en tvådimensionell matris, en rad för varje kammare och en kolumn för varje datatyp. Kallas en gång per mättillfälle.
Check static clust	Bygger ett första statusmeddelande till frontpanelen, figur 11, vilket visar eventuella felmeddelanden i respektive kammare samt status för turbopumparna. Kallas en gång när huvudprogrammet startas.
Data array size	Räknar antalet kammare samt maximala antalet datatyper per mätserie, vilket bland annat används för att definiera storleken på matrisen som skapas i Data all chamb.
Init Config	Öppnar kommunikationen med samtliga tryckmätare och turbopumpar.
Chamb. Data	Hämtar tryck- och turbopumpsdata från en specificerad kammare. Datan skickas till Data all chamb.
Merge Error	Sammanfogar två felmeddelanden till ett.
Vac const	Håller tröskelvärden för HV och LV. Dessa används för att avgöra om det är säkert att starta turbopumpar samt för att indikera vilka kammare som uppnått HV och LV på frontpanelen, markering 4 i figur 11.
Config Visa Com Port	Öppnar kommunikation med en turbopump eller tryckmätare enligt parametrarna i tabell 4.
Get pressure	Avläser ett tryckvärde från en tryckmätare.
Error Code Pr	Översätter eventuell felkod skickad från en tryckmätare till klartext. Texten skickas tillsammans med felkod -8002 av Get Pressure.
Write And read	Skickar ett kommando och läser av svaret från en tryckmätare.
TD 400 Start + Data	Kontrollerar en turbopump; skickar ett kommando och avläser svaret.
Wait 4 Data	Väntar på att data ska skickas från ett mätinstrument till datorn.
Error TD 400	Avläser eventuell felkod skickad från en turbopump och översätter den till klartext. Texten skickas tillsammans med felkod -8004 från TD 400 Start + Data.



New Error	Sammanfogar ett booleskt värde, en felkod och ett felmeddelande till ett felkluster.
-----------	--

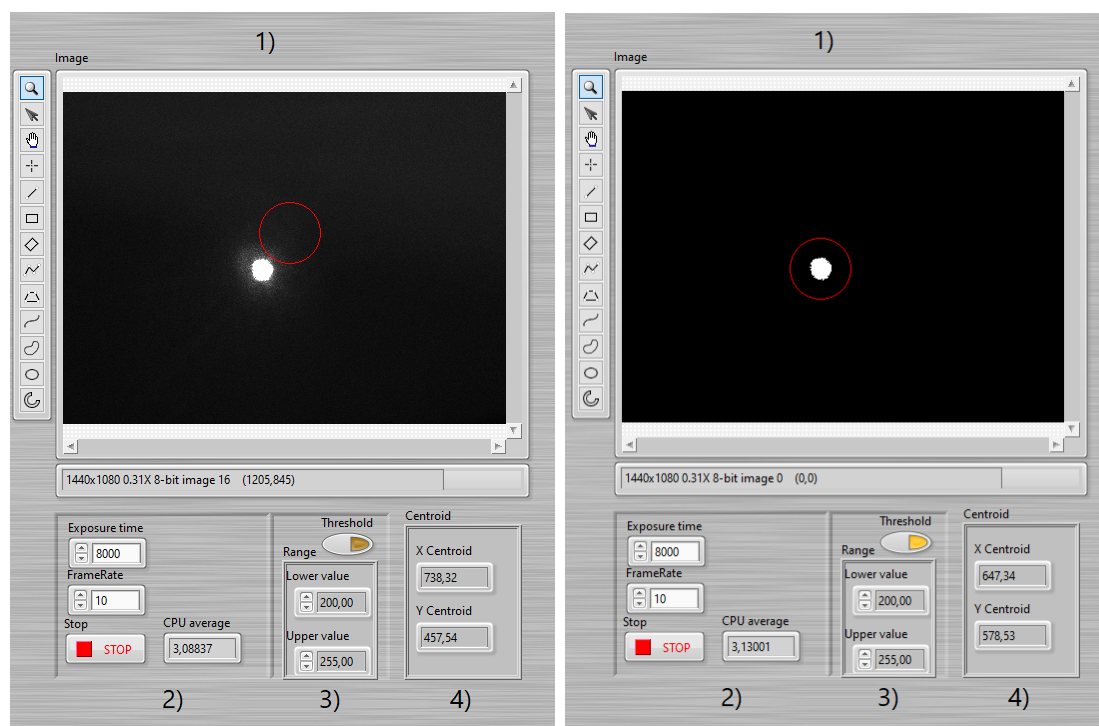
Tryckprogrammet har för närvarande ett känt problem. Det visar sig att felkod '-8003' returneras och programmet slutar kommunicera med turbopumpen. Pumpen är fortfarande aktiv och fungerande men problemet som uppstår är att inga värden uppdateras i graferna för turbopumpen. Felkod '-8003' betyder att fel antal bytes returneras. Anledningen till problemet är att när felmeddelandet '-8003' skickades genom 'VISA read' i 'Wait 4 Data' stoppas den biten av programmet och hoppar över hela avläsningen. När en ny iteration påbörjas ligger därmed all data från den tidigare iterationen kvar, vilket medför att samma felkod returneras igen. Lösningen som har implementerats är att alltid kalla 'VISA read', oberoende om ett fel uppkommit eller inte, för att nollställa informationen från turbopumpen. Vid tillfällen då för få bytes skickas under en iteration sätts värden från avsaknad information enbart till värdet noll. Ett kvarstående problem är att enbart 15 bytes fortfarande returneras vid vissa tillfällen. Vi vet inte varför detta sker. Med största sannolikhet är det inte från vårt program utan något utomstående, då konstant information skickas in. Detta fel har uppkommit mer sällan än en gång per timme vid testkörningar av programmet och innebär därmed inga större hinder för användandet av programmet, då data hämtas från tryckinstrumenten varje sekund.

## 5.2 Optik

Resultatet av styrsystemet för kamerorna representeras i figur 16 som visar frontpanelen för LabVIEW-programmet. Styrsystemet för kameran består av en while-loop. Utanför while-loopen startas kommunikationen med en vald kamera och datorns processor, som var Intel Core i7-10700. Därefter konfigureras kameran för att kontinuerligt hämta bilder istället för att endast hämta en bild. Med denna konfigurering kunde vi använda en while-loop där en iteration utfördes varje gång en ny bild hämtades från kameran. Inuti while-loopen hämtade vi data för processoranvändningen (markering 2), bildinformationen från kameran, beräknade laserns position (markering 4) samt visade bilden på frontpanelen (markering 1). Optimering för beräkningen av laserns position möjliggjordes genom användning av ett tröskelvärde, se markering 3, där *lower value* motsvarar *thValue* beskrivet i avsnitt 4.2.2. Positionsbestämningen för lasern med och utan tröskelvärdet visas i figur 16b respektive 16a där det tydligt kan ses att användningen av tröskelvärdet ger en bättre positionsbestämning. Möjligheten att anpassa kameraattribut infördes så att användaren kan ändra kamerans exponeringstid samt bildfrekvens, se markering 2. En lista på namn och förklaringar för de sub:VIs vi skapat finns i tabell 10.

I tabell 11 redvisas den totala processorkapaciteten som använts vid körning av programmet samt överföringshastigheten av information från kameran. Detta är viktigt då piezomotorerna senare ska positionera speglarna efter beräkningen av respektive kamerabilds masscentrum. Utifrån de erhållna värdena för total processoranvändning, samt under antagandet att processoranvändningen ökar linjärt med antalet kameror, kan det konstateras att den använda datorn bör vara kapabel att kontrollera samtliga sex kameror markerade i figur 1. Detta skulle kräva att datoruppställningen är kapabel till en

överföringshastighet i storleksordningen 1 GB/s.



(a) Frontpanelen utan tröskelvärdesfunktion aktiv  
(b) Frontpanelen med tröskelvärdesfunktion aktiv

Figur 16: Frontpanelen för LabVIEW-programmet för kameran. 1) Kamerans bild i realtid. 2) Inställning av kamerans attribut. Genomsnittliga användandet av processorn. 3) På- och av-knapp samt insättning av tröskelvärde. 4) Bildens masscentrum för x- respektive y-koordinat. Origo är placerat längst upp i vänstra hörnet av bilden.

Tabell 10: Namn och förklaringar för sub:VIs vi skapat för optikprogrammet.

Sub:VIs	Förklaring
Set cam att	Ställer in kamerans attribut som valts på frontpanelen.
CPU init	Startar kommunikationen med datorns processor.
CPU data	Hämtar den procentuella användningen av processorn.
Find CM	Anger tröskelvärden som används för beräkningen av bildens masscentrum samt markerar masscentrum med en röd ring i frontpanelens bild.

Tabell 11: Total processoranvändning och överföringshastighet mellan kamera och dator. Processorn som användes vid mätning var Intel Core i7-10700.

Bildfrekvens [Hz]	Total processoranvändning [%]	Överföringshastighet [MB/s]
10	2.91	16
30	6.09	47
50	10.5	79
70	9.74	110

Vidareutveckling för detta LabVIEW-program krävs för att uppnå automatisering. De områden som har utelämnats är kommunikation med piezomotorn och sammankopplingen av kamera och piezomotorn. Sammankopplingen mellan dessa system kan lösas genom att definiera piezomotorns rotation med ändringen av kamerabildens masscentrum. Detta skulle då automatisera positioneringen av spegeln inuti vakuumkammaren. Optimering av programmet kan utföras för att minska processoranvändningen.

## 6 Slutsats

Stora delar av det mål som sattes för arbetet, att skapa ett styrsystem för HHG-uppställningen i Attohallen vid Göteborgs universitet, har uppnåtts. Tryckregleringssystemet kan kontinuerligt hämta tryckdata och styra turbopumpar, vilka är kapabla att producera ett HV i de testkammare som använts. Vidare krävs relativt små modifikationer i koden för att styra fler tryckmätare och pumpar. Optikdelen av styrsystemet är kapabelt till att hämta bilder från en kamera, bestämma kamerans exponeringstid och bildfrekvens, samt avgöra positionen för en laser i bilden. Den återstående delen av styrsystemet som behöver utvecklas är kommunikationen med, och den automatiserade kontrollen av, piezomotorerna. Vidare bör det poängteras att mjukvaran ej har kunnat testas i den slutgiltiga HHG-uppställningen.

Sammanfattningsvis kan LabVIEW-programmet för tryckregleringen användas till Attohallen vid Göteborgs universitet, vilket vi hoppas på sikt kan bidra till en djupare förståelse om atomära och molekylära egenskaper.

## 7 Referenser

- [1] C. Arnold, M. Isinger, D. Busto, D. Guénot, S. Nandi, S. Zhong, J. Dahlström, M. Gisselbrecht och A. IHuillier, “How can attosecond pulse train interferometry interrogate electron dynamics?”, English, *Photoniques*, s. 28–35, 2018, ISSN: 1629-4475. DOI: [10.1051/photon/2018S328](https://doi.org/10.1051/photon/2018S328).
- [2] D. Busto, *Quantum interference effects in attosecond photoionization dynamics [Elektronisk resurs]*. Lund: Department of Physics, Lund University, 2020, ISBN: 9789178955602.
- [3] J. Solis, J. Siegel, C. Afonso, N. Barry, R. Mellish och P. French, “Experimental study of a self-starting kerr-lens mode-locked titanium-doped sapphire laser”, *Optics Communications*, årg. 123, nr 4, s. 547–552, 1996, ISSN: 0030-4018. DOI: [https://doi.org/10.1016/0030-4018\(95\)00578-1](https://doi.org/10.1016/0030-4018(95)00578-1). URL: <https://www.sciencedirect.com/science/article/pii/0030401895005781>.
- [4] M. Grünig, F. Staub och J. Balmer, “Nd:glass regenerative amplifier for cpa (chirped-pulse amplification) lasers”, *Optics and Lasers in Engineering*, årg. 43, nr 3, s. 263–269, 2005, Optics in Switzerland, ISSN: 0143-8166. DOI: <https://doi.org/10.1016/j.optlaseng.2004.04.006>. URL: <https://www.sciencedirect.com/science/article/pii/S0143816604000909>.
- [5] H. Coudert-Alteirac, *Spatial and temporal metrology of intense attosecond pulses [Elektronisk resurs]*. Lund: Department of Physics, Lund University, 2018, ISBN: 9789177537854.
- [6] A. A. Voronin och A. M. Zheltikov, “The generalized sellmeier equation for air”, *Scientific Reports*, årg. 7, nr 1, s. 46 111, aug. 2017, ISSN: 2045-2322. DOI: [10.1038/srep46111](https://doi.org/10.1038/srep46111). URL: <https://doi.org/10.1038/srep46111>.
- [7] E. Gullikson, “X-ray interactions with matter”, 2010. URL: [https://henke.lbl.gov/optical\\_constants/](https://henke.lbl.gov/optical_constants/) (hämtad 2021-04-28).
- [8] J. H. I. Thomas, D. M. Hoffman och B. Singh, *Handbook of Vacuum Science and Technology [Elektronisk resurs]*. San Diego: Academic Press, 1998.
- [9] Sens4 A/S, “What is the working principle of the pirani gauge?”, URL: <https://sens4.com/pirani-working-principle.html> (hämtad 2021-03-25).
- [10] Pfeiffer Vacuum GmbH, “Indirect, gas-dependent pressure measurement”, URL: <https://www.pfeiffer-vacuum.com/en/known-how/vacuum-measuring-equipment/fundamentals-of-total-pressure-measurement/indirect-gas-dependent-pressure-measurement/> (hämtad 2021-03-25).
- [11] APC, “What is the purpose of a piezo motor?”, mars 2020. URL: <https://www.americanpiezo.com/blog/what-is-the-purpose-of-a-piezo-motor/> (hämtad 2021-04-27).
- [12] D. Reynders, S. Mackay, E. Wright och S. Mackay, “4a - rs-232 overview”, i *Practical Industrial Data Communications*, D. Reynders, S. Mackay, E. Wright och S. Mackay, utg., Oxford: Butterworth-Heinemann, 2004, s. 71–81, ISBN: 978-0-7506-6395-3. DOI: <https://doi.org/10.1016/B978-075066395-3/50004-8>. URL: <https://www.sciencedirect.com/science/article/pii/B9780750663953500048>.
- [13] *Serial interfaces for turbo.drives, l, 300, 400 and 700*, Oerlikon Leybold Vacuum, 2008.

## A Textrepresentationer av numeriska värden

Textsymboler med tillhörande decimalt värde

33: !	34: "	35: #	36: \$	37: %	38: &	39: '	40: (	41: )	42: *	43: +	44: ,	45: -	46: .	47: /	48: 0
49: 1	50: 2	51: 3	52: 4	53: 5	54: 6	55: 7	56: 8	57: 9	58: :	59: ;	60: <	61: =	62: >	63: ?	64: @
65: A	66: B	67: C	68: D	69: E	70: F	71: G	72: H	73: I	74: J	75: K	76: L	77: M	78: N	79: O	80: P
81: Q	82: R	83: S	84: T	85: U	86: V	87: W	88: X	89: Y	90: Z	91: [	92: \	93: ]	94: ^	95: _	96: `
97: a	98: b	99: c	100: d	101: e	102: f	103: g	104: h	105: i	106: j	107: k	108: l	109: m	110: n	111: o	112: p
113: q	114: r	115: s	116: t	117: u	118: v	119: w	120: x	121: y	122: z	123: {	124:	125: }	126: ~	127: ¨	128: €
129: ¸	130: ,	131: f	132: „	133: …	134: +	135: ‡	136: °	137: ‰	138: Š	139: ¢	140: Œ	141:	142: Ž	143:	144:
145: ´	146: ´	147: ¨	148: ¨	149: •	150: –	151: —	152: ~	153: ™	154: š	155: ›	156: œ	157:	158: ž	159: Ÿ	160:
161: ¡	162: ¢	163: £	164: ¤	165: ¥	166: ¦	167: §	168: ¨	169: ©	170: º	171: «	172: ¬	173: -	174: ®	175: ¯	176: °
177: ±	178: ²	179: ³	180: ´	181: µ	182: ¶	183: ·	184: ¸	185: ¹	186: º	187: »	188: ¼	189: ½	190: ¾	191: ¿	192: À
193: Á	194: Â	195: Ã	196: Ä	197: Å	198: Æ	199: Ç	200: È	201: É	202: Ê	203: Ë	204: Ì	205: Í	206: Î	207: Ï	208: Ð
209: Ñ	210: Ò	211: Ó	212: Ô	213: Õ	214: Ö	215: ×	216: Ø	217: Ù	218: Ú	219: Û	220: Ü	221: Ý	222: Þ	223: ß	224: à
225: á	226: â	227: ã	228: ä	229: å	230: æ	231: ç	232: è	233: é	234: ê	235: ë	236: ì	237: í	238: î	239: ï	240: ð
241: ñ	242: ò	243: ó	244: ô	245: õ	246: ö	247: ÷	248: ø	249: ù	250: ú	251: û	252: ü	253: ý	254: þ	255: ÿ	

Figur 17: Textsymboler som används i LabVIEW med tillhörande decimalt värde. Värden 0 till och med 32 samt 127 kodar för olika kommandon, mellanslag och dylikt, vilka saknar teckenrepresentationer.

Vid kommunikation med instrument genom LabVIEW skickas data som en textsträng<sup>1</sup> uppbyggd av symbolerna i figur 17. Kommandon med decimala representationer mellan 0 och 32 kan exempelvis hittas på wikipedia<sup>2</sup>.

<sup>1</sup>OBS: 77 106 117 107 118 97 114 97 110 32 112 114 101 115 101 110 116 101 114 97 116 32 105 32 100 101 116 116 97 32 97 114 98 101 116 101 32 228 114 32 100 101 115 105 103 110 97 100 32 102 246 114 32 97 116 116 32 115 107 106 117 116 97 32 108 97 115 101 114 112 117 108 115 101 114 32 109 101 100 32 104 246 103 32 105 110 116 101 110 115 105 116 101 116 32 105 32 118 97 107 117 117 109 46 10 83 116 117 100 101 110 116 101 114 110 97 32 115 111 109 32 109 101 100 118 101 114 107 97 116 32 105 32 100 101 115 105 103 110 101 110 32 97 118 32 109 106 117 107 118 97 114 97 110 32 97 118 115 228 103 101 114 32 115 105 103 32 97 110 115 118 97 114 101 116 32 111 109 32 100 101 110 32 97 110 118 228 110 100 115 32 102 246 114 32 97 116 116 32 102 246 114 115 116 246 114 97 10 112 108 97 110 101 116 101 114 46 32

<sup>2</sup>[https://en.wikipedia.org/wiki/ASCII#Control\\_code\\_chart](https://en.wikipedia.org/wiki/ASCII#Control_code_chart)

## B Fullständig kod

Alla LabVIEW-program för både vakuumbildning och optik kan hämtas via:

<https://drive.google.com/drive/folders/15fJ8nKiy9xgUDNK5DuYdnHvRqCpWYZ8?usp=sharing>



GÖTEBORGS  
UNIVERSITET

---



**CHALMERS**